

DOCTORAL THESIS

Robust Web Annotations in Support of Knowledge Co-Creation

Vishwajeet Pattanaik

TALLINN UNIVERSITY OF TECHNOLOGY
DOCTORAL THESIS
40/2022

Robust Web Annotations in Support of Knowledge Co-Creation

VISHWAJEET PATTANAİK



TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies
Department of Software Science

**The dissertation was accepted for the defence of the degree of Doctor of Philosophy
on 13 June 2022**

Supervisor: Prof. Dr. Dirk Draheim,
Information Systems Group,
Department of Software Science,
School of Information Technologies,
Tallinn University of Technology
Tallinn, Estonia

Opponents: Prof. Dr. Peter Thiemann,
Department of Computer Science,
University of Freiburg,
Freiburg, Germany

Assoc.-Prof. Dr. Ismail Khalil,
Institute of Telecooperation,
Johannes Kepler University Linz,
Linz, Austria

Defence of the thesis: 27 July 2022, Tallinn

Declaration:

Hereby I declare that this doctoral thesis, my original investigation and achievement, submitted for the doctoral degree at Tallinn University of Technology, has not been submitted for any academic degree elsewhere.

Vishwajeet Pattanaik

signature

Copyright: Vishwajeet Pattanaik, 2022
ISSN 2585-6898 (publication)
ISBN 978-9949-83-868-4 (publication)
ISSN 2585-6901 (PDF)
ISBN 978-9949-83-869-1 (PDF)
Printed by Auratrükk

TALLINNA TEHNIKAÜLIKOOL
DOKTORITÖÖ
40/2022

Töökindlad veebiannotatsioonid teadmiste ühisloome toetamiseks

VISHWAJEET PATTANAİK



Contents

List of Publications	8
Author's Contributions to the Publications	9
Abbreviations.....	10
1 Introduction	11
1.1 Problem Statement	13
1.2 Research Questions	14
1.3 Research Methodology	15
1.4 Contributions	15
1.5 Dissertation Outline.....	16
2 Background and Related Work	17
2.1 Annotations	17
2.2 Web Annotation Systems	18
2.2.1 Web Annotation Systems before 2010	18
2.2.2 Hypothes.is	19
2.2.3 Web Annotation Data Model	20
2.3 Annotations as a Means for Knowledge Co-Creation.....	21
2.4 Link Rot and Web Page Decay	22
2.5 Losing Annotations.....	26
3 Web Annotations as a Tool in Support of the Knowledge Management Lifecycle .	27
3.1 Overview of Frameworks for Knowledge Transformation.....	27
3.2 Overview of Nonaka's SECI Model.....	28
3.3 Web Annotation Activities	30
3.3.1 Semantic Annotations.....	31
3.4 SECI through Annotation Activities	32
3.4.1 Socialization Activities.....	32
3.4.2 Externalization Activities	34
3.4.3 Combination Activities	35
3.4.4 Internalization Activities	36
3.5 Discussion and Conclusions.....	36
4 A Novel Anchoring Algorithm for Textual Web Annotation	39
4.1 State-of-the-Art	39
4.1.1 Anchoring Annotations using Keywords.....	41
4.1.2 The Fuzzy Anchoring Algorithm.....	42
4.2 The Novel Anchoring Algorithm	44
4.2.1 The DOM-Oriented Edit-Distance Algorithm	44
4.2.2 Experimental Setup and Results	48
4.2.3 Discussion.....	50
4.3 Summary and Conclusions.....	51
5 The Web Annotation Platform Tippanee	52
5.1 Conventional Web Annotation Activities.....	52
5.2 Artifact Development	54
5.3 Novel System Features	56

5.3.1	Local Storage	56
5.3.2	Reconstructing Anchors	56
5.3.3	Similarity Index	57
5.3.4	Linking Annotations	57
5.3.5	Transclusions	57
5.4	Lab Experiment	58
5.5	Results and Discussion	59
5.6	Summary and Conclusions	63
6	Evaluating the Robustness of Anchoring Algorithms	65
6.1	Overview of the Web Annotation Test Bench	65
6.2	Test Bench Setup	66
6.2.1	Creating Annotations	67
6.2.2	Simulating Web Page Decay	68
6.2.2.1	Textual Decay Types	68
6.2.2.2	Structural Decay Types	69
6.3	Results and Discussion	71
6.4	Summary and Conclusions	73
7	Conclusions	80
7.1	Summary of Thesis Contributions	80
7.2	Future Directions	82
7.2.1	Adding a Layer of Security Through Peer-to-Peer	82
7.2.2	Guidelines for Online Collaborative Communities	83
7.2.3	Annotations in Support of an Open, Democratic Web	83
	List of Figures	85
	List of Tables	86
	List of Listings	87
	References	88
	Acknowledgements	101
	Abstract	102
	Kokkuvõte	104
	Appendix 1	107
	Appendix 2	117
	Appendix 3	129
	Appendix 4	151
	Appendix 5	163
	Appendix 6	201

Appendix 7 - Getting Started Guide for Tippanee	213
Curriculum Vitae	225
Elulookirjeldus.....	229

List of Publications

The present Ph.D. thesis is based on the following publications that are referred to in the text by Roman numbers.

- I V. Pattanaik, A. Norta, M. Felderer, and D. Draheim. Systematic support for full knowledge management lifecycle by advanced semantic annotation across information system boundaries. In J. Mendling and H. Mouratidis, editors, *Information Systems in the Big Data Era*, pages 66–73, Cham, 2018. Springer International Publishing
- II V. Pattanaik, S. Suran, and D. Draheim. Enabling social information exchange via dynamically robust annotations. In *Proceedings of the 21st International Conference on Information Integration and Web-Based Applications & Services*, iiWAS2019, page 176–184, New York, NY, USA, 2019. Association for Computing Machinery
- III V. Pattanaik, I. Sharvadze, and D. Draheim. Framework for peer-to-peer data sharing over web browsers. In T. K. Dang, J. Küng, M. Takizawa, and S. H. Bui, editors, *Future Data and Security Engineering*, pages 207–225, Cham, 2019. Springer International Publishing
- IV V. Pattanaik, I. Sharvadze, and D. Draheim. A peer-to-peer data sharing framework for web browsers. *SN Computer Science*, 1(4), June 2020
- V S. Suran, V. Pattanaik, and D. Draheim. Frameworks for collective intelligence: A systematic literature review. *ACM Comput. Surv.*, 53(1), Feb. 2020
- VI S. A. Peious, S. Suran, V. Pattanaik, and D. Draheim. Enabling sensemaking and trust in communities: An organizational perspective. In *Proceedings of the 23rd International Conference on Information Integration and Web-Based Applications & Services*, iiWAS '21, page 1–9, New York, NY, USA, 2021. Association for Computing Machinery

Author's Contributions to the Publications

- I In **publication I**, I was the main author; I defined the research problems, and conducted the background study. I developed the framework, prepared the figures, and wrote the manuscript.
- II In **publication II**, I was the main author and defined the research problems. I developed the algorithm, wrote the code and implemented the web annotation system. I planned for and conducted the lab experiment, analysed the results, prepared the figures, and wrote the manuscript.
- III In **publication III**, I was the main author; I defined the research problems. I assisted in the development of the app, and wrote the manuscript.
- IV In **publication IV**, I was the main author; I defined the research problems. Analysed the results of the lab experiment, prepared the figures, and wrote the manuscript.
- V In **publication V**, I was a co-author; I contributed to the development of the research methodology. I cross-verified the selected studies, and assisted in the development of the framework. I contributed to the discussion in the paper.
- VI In **publication VI**, I was a co-author and contributed in the formulation of the research questions. I revised and improved the manuscript.

Abbreviations

API	Application Programming Interface
CA	Cambridge Analytica
CI	Collective intelligence
CSCW	Computer-Supported Cooperative Work
CSS	Cascading Style Sheets
DB	Database
DOM	Document Object Model
ERP	Enterprise Resource Planning
GDPR	General Data Protection Regulation
HTML	HyperText Markup Language
ICT	Information and Communications Technology
IS	Information Systems
JS	JavaScript
JSON	JavaScript Object Notation
OAD	Open Annotation Data
P2P	Peer-to-peer
SECI	Socialization, Externalization, Combination, and Internalization
SVG	Scalable Vector Graphics
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WAD	Web Annotation Data
WWW	World Wide Web
XML	eXtensible Markup Language
XPath	XML Path

1 Introduction

The Web today, through its plethora of technologies and applications, empowers its users by enabling interactions, communications, and collaborations on a global scale. One of these applications, that has been of particular importance for knowledge sharing and learning, and has existed since the inception of the web, is web annotation. Unfortunately, although web annotation systems have enabled users to exchange and critique pieces of (textual) web content; challenges being faced on the web (in the recent decade), namely: difficulty in identifying, finding, evaluating, and using information effectively, volatility of web contents, and the lack of innovation in web annotation technologies have together prevented these systems from becoming a primary tool for online learning and knowledge sharing. To deal with these challenges, the work presented in this dissertation proposes a novel strategy for the detection and reattachment of web annotations. The proposed strategy (i.e., novel anchoring algorithm) is designed into a novel browser-based web annotation platform that comes with its own new set of system features and is empirically validated against the current state-of-the-art in web annotation systems through a first of its kind annotation test-bench. The significance of the challenges this dissertation addresses can be better gauged by delving into the impact web technologies have had on our societies since its inception and by examining the state, the web is in right now.

The World Wide Web (WWW), as it was conceived three decades ago, was primarily designed to serve as a *universal linked information system* [36, 35] that enables flow of information between different computer systems [37, 39] through the use of *hypertext* [127, 128]. Since then, communities on the web (together with researchers and academicians) have produced, through creativity and collaboration, several ground-breaking ideas and innovations. These innovations have allowed the web to transcend from a mere medium for broadcasting, to a dynamic social environment. This transformation has enabled the web to emerge as a predominant mechanism for global communication and collaboration in a wide variety of domains, ranging from education and science, to economy, democracy, culture, and more [113, 38]; allowing for paradigm shifts in problem-solving [88, 119], knowledge-exchange [67, 166], and innovation [66, 74] – in general, enabling mobilizations of skills and knowledge in unprecedented ways [114, 119].

Unfortunately, this unmitigated enthusiasm around the applications (and benefits) of the Internet (and the web), has started to “wane in the face of a *nasty storm* of issues” [170] over the years [154]. Issues such as bots, clickbait, filter bubbles [140], echo chambers [90, 54], fake news [109, 29], and ideological polarization [64] now present a new constant threat to online collectives (i.e., web users) [123], so much, that many experts fear that disruptions caused by use of technologies will “mostly weaken core aspects of democracy” in the next decade [154]. Although the impact of some of these issues have already been brought to light, for example by reports on the Cambridge Analytica (CA) scandal [165, 34, 154], or the 2016 EU Referendum (“Brexit”) [77, 154], or the more recent 2021 Capitol riots [84]; the (larger) impact of the above mentioned issues are only now being deeply examined in scientific literature. For example, how these issues are influencing citizens’ collective intelligence and behaviour, and what needs to be done to prevent said influence [62, 180, 117, 110, 93, 28].

A second distinct challenge that has become increasingly worrisome is the expansive and volatile nature of today’s web [55]. Although, concerns around the ephemerality of content on the web has been discussed in detail in literature over the years [134, 167, 22], the advent of social media (enabled by Web 2.0 technologies [138]) has made the situation only worse. As stated by Schneider et al. [167], web content is inherently ephemeral in its construction, just like television or other forms of media. Once presented, the content

needs to be deconstructed and re-constructed into various forms for users to experience it on different (forms of) media. Say for example, programming tutorials on YouTube, that are often re-uploaded to various video sharing platforms, or are converted (i.e., deconstructed and re-constructed) into textual form to be shared on online publishing platforms like Medium, or added to GitHub as code. This ephemerality of web content is further made worse by the fact that online content can only be expected to be visible for a brief period of time [167]. In part thanks to search engines (on the web) and recommendation systems (on social media platforms) that encourage web users' short attention spans [59]. Also, as more citizens around the world get access to the Internet and sign-up to the numerous social media platforms available on the web, more and more content is produced, thereby further exacerbating the ephemeral nature of web content. It has been estimated that in 2015-2017, web users (and services) alone have generated more data than what was created in "all of the preceding history" [168]. This is disconcerting as researchers fear that "all this information will likely vanish in a few years, creating a knowledge gap about the present for future generations" [55]. Also, this *decay* or *rot* of knowledge has serious implications for science and education (and knowledge exchange in general), which has been brought up in literature by several researchers [22, 153]. For example, as illustrated by Klein et al. in their study [102], where they analysed more than 3.5 million scholarly articles containing 1 million Uniform Resource Identifier (URI) references to *web-at-large* resources, and found that 65% of cited URIs lacked any Mementos (i.e., the cited references were not achieved); and this indeed entailed that the cited content "might no longer reflect the citing author's original intent" [153] and that the readers may be oblivious to this change.

That said, as can be seen from the thriving literature on detection of fake news [192], role of bots on social media [48], effect of echo chambers [92] and more, it is clear that researchers have developed several solutions to tackle many of the above mentioned issues. A first common theme that emerges from many of the proposed solutions is to leverage the *wisdom of the crowd* (i.e., the collection intelligence (CI) [113]) through collaborative processes such as crowdsourcing (for example as illustrated by [169]). CI, which, in the field of information and communications technology (ICT), is understood as "universally distributed intelligence, constantly enhanced, coordinated in real time, and resulting in the effective mobilization of skills" [113]. The topic has recently gained tremendous interest in research and development, as can be gauged by the numerous crowd-oriented CI systems being developed and used by both governments and organizations world-wide. The second theme (of solutions) that has emerged more recently is the emphasis on *media and information literacy* [91, 94]. Here, *media literacy* encompasses practices that enable an individual to access, create, analyze, and evaluate any form of media [157]; while *information literacy* emphasizes on an individual's ability to navigate and locate reliable information [87]. It is critical to note here that literature suggests that although media literacy can play a critical role in identification of fake news [72, 164, 78], in practice, information literacy is more relevant for fake news recognition [94].

As for the issues of *web page decay* [30] and *reference rot* (defined as a combination of *link rot* and *content drift*) [102], a plethora of web archival services (like: Perma.cc, WebCitation, Archive.today, Wayback Machine, UK Web Archive and Memento) that enable preservation of digital information published on the web, have been developed over the years [55]. However, even with web archiving gaining recognition from both general web users and research communities, almost all web archival initiatives "exclusively hold content related to their hosting country, region or institution" [55]; often due to financial limitations, exacerbated by the expansive nature of web content. Furthermore, there is

the technology aspect. As reported by Costa et al. [55], users find it difficult to search through full texts of archived web pages, and the provided search tools often result in unsatisfactory results. Unfortunately, both issues are technological issues (i.e., difficult to implement), and the ephemerality of web content makes it only more difficult to design better search technologies [55].

With these issues, challenges, and emerging solutions in mind, this dissertation explores how web users can be empowered in such a way that they are able to find, organize, use, evaluate and communicate information, while not having to fully rely on content and service providers. In particular, we develop a user-centric, crowd-oriented solution in support of media and information literacy, that mitigates the current gaps in technology.

1.1 Problem Statement

Among the wide variety of solutions available today in support of media and information literacy (i.e., knowledge and information exchange), a common class of *computer-supported collaborative work* (CSCW) tools that have reemerged in research recently, especially in e-learning communities, are *web annotations*. Annotations are pieces of information, typically explanations and comments, that are added to specific parts of media (conventionally text); for example, scribbles, doodles or highlights on books. In context of the web, annotations were first introduced in the *Mosaic Browser* in 1993 [104] and since then have been recognized as a fundamental notion of hypertext systems [121]. Defined as “lightweight, efficient, non-intrusive (preferably transparent), platform-independent and scale-able” systems, web annotations are meant to allow third-parties (i.e., non-owners of web content) to interact and *incrementally* augment web documents [104], allowing readers to “intelligently contextualize” documents based on their interests [104]. However, over the past decades, several web annotation systems have been developed, launched and then discontinued; many of which were owned commercially. Among these web annotation systems, the ones still in use today include Diigo [11], A.nnotate [2], Genius [4] and Hypothes.is [7]. These web annotation systems are designed as web-based applications and platforms that allow their users to create, organize and share textual web annotations; thereby enabling annotations to act as “a conversation layer over the entire web” [7]. Most of these applications can be used by browsing to the URIs of these platforms; while in some cases these applications come with extensions (or apps) that can be installed by users within their browsers – allowing web users to interact with web content irrespective of any web pages’ underlying heterogeneous technologies. Researchers, however, have argued that, given that most of these web annotation systems are privately owned, the annotations created on these platforms often end up restricted in their application silos [53]. Thereby compelling the users of these web annotation platforms to rely on their service providers, for services such as storage, retrieval, and organization of the annotations that are originally created by the users themselves. Thus preventing the users from having full control of their data (specifically, their web annotations).

Considering this, the World Wide Web Consortium (W3C) in 2017, proposed the Web Annotation Data (WAD) model as an alternative solution for enabling web annotations. The new WAD model provides a “standard description model and format to enable annotations to be shared between systems” [13]. Designed to serve as an interoperable and extensible framework for expressing annotations on the web, the WAD model aims to treat web annotations as “first-class objects” [53]. Unfortunately, because the responsibility of integrating the WAD model into websites lies with the content providers, the WAD model has yet to be widely adopted by developers and the web in general. On the other hand, web annotation systems like *Hypothes.is* have been garnering tremendous

support from web communities; so much so, that they reached their 20 millionth annotation in 2021, twice as much as the previous year, and four times as much as the year before that [9]. A key issue however with *Hypothes.is* (and other similar systems) is that, these systems are unable to keep up with the ephemerality of web content. As discovered by Aturban et al., 27% of the highlighted text annotations created on *Hypothes.is* are already orphaned [26]; that is to say that 27% of the highlighted text annotations studied by the authors, could not be reattached to their original web pages. The authors also found that further 61% of the studied annotations were at risk of being orphaned [26], and the primary reason for this was the decay of content on the annotated web pages. Other web annotation systems with less users suffer from the decay issue, too, as all of these systems rely on simple *keyword anchoring*; a technique that was first proposed in 2001 [45]. The approach searches through an annotated documents and attempts to find the keywords (i.e., textual strings) to which a highlight or comment was originally anchored (i.e., attached), then, if the keywords (or, in most cases some part of the keywords) are found, the associated highlight or comment is attached to that part of the document [45]. However, as mentioned previously, this can become more and more difficult over time as web pages keep decaying.

All of this entails that, although the technological solutions required for enabling crowd-centric knowledge and information sharing are already available in literature and in practice; these solutions either receive less support from the community or these solutions are not able to keep up with the numerous challenges being observed on the web. To tackle these issues and gaps in technology, the work presented in this dissertation proposes several new strategies that build upon emerging ideas, and addresses the critical challenge of enabling *information literacy* in presence of *web page decay*. This dissertation explores system features and tool sets that are essential for enabling knowledge and information sharing, when using web annotation systems as *conversation layer* over the heterogeneous web.

1.2 Research Questions

To summarize, the overarching goal of this dissertation is to improve the state of the art in web annotation systems in support of media and information literacy. This is achieved by addressing two primary research questions (RQs); subsequently broken down into three sub-RQs each. As outcome of the answering the research questions, the dissertation makes four distinct contributions C1-C4 that will be explained in due course in Sect. 1.4. The relevant associations between the RQs, dissertation chapters, and contributions are summarized in Table 1.

RQ1 How to support knowledge creation and sharing through use of web annotations?

RQ1.1 What insights from knowledge management can be incorporated into web annotation processes?

RQ1.2 How can web annotations support knowledge management life cycles?

RQ1.3 What system features have to be fulfilled by solutions to support knowledge creation and sharing, when using web annotations?

RQ2 How to make textual web annotations robust against ephemerality of web content?

RQ2.1 How can web annotations be prevented from being orphaned?

RQ2.2 How to design a stable web annotation platform that does not fully rely on content providers?

RQ2.3 How to evaluate the robustness of anchoring algorithms?

Table 1 – Research questions with associated dissertation chapters and contributions.

Research Questions	Chapter	Contribution
RQ1.1, RQ1.2, RQ1.3	3	C1
RQ2.1	4	C2
RQ2.2	5	C3
RQ2.3	6	C4

1.3 Research Methodology

To answer the above mentioned RQs, the dissertation employs methodologies from Information Systems (IS) research. In particular, the work presented in the dissertation is rooted in the best practices and principles of high-quality design science research [83]. The contributions presented in this dissertation are four distinct artifacts, each intended to solve the previously laid out challenges and technological gaps. Following the principles of design science, the contributions made in this work are evaluated using three distinct methodologies, namely, *Informed Arguments*, *Controlled Experiments*, and *Simulations* [83]. The evaluation methodologies used in this dissertation corresponding to the respective contributions are illustrated in Table 2.

Table 2 – Mapping of dissertation contributions, proposed artifacts and corresponding evaluation methodologies.

Contribution	Artifact	Evaluation Methodology
C1	KM framework for annotation activities	Informed Arguments
C2	Novel anchoring algorithm	Controlled Experiment
C3	Web annotation system	Controlled Experiment
C4	Web annotation test-bench	Simulations

1.4 Contributions

The dissertation develops both conceptual and technological solutions that enable web users to create and share knowledge and information through textual annotations on the web, not only individually but also collectively. With web users' requirements at its core, this dissertation makes four distinct contributions to research in knowledge management and web annotation systems:

- C1** A framework for web annotation activities in support of knowledge management life cycles. The proposed framework draws influence from Nonaka's knowledge spiral that explains creation of new knowledge in organizational settings by converting tacit knowledge into explicit knowledge, and vice versa.
- C2** A novel anchoring algorithm that is robust with respect to ephemeral web content, and is thereby able to reattach textual annotations to their appropriate anchors even when an underlying anchor has decayed. The proposed anchoring algorithm is evaluated through a controlled experiment, in which the algorithm's robustness is compared to the current state of the art.

- C3** The design and development of a stable web annotation system that utilizes the proposed anchoring algorithm (C2). The proposed web annotation system, enabled by its anchoring algorithm, allows its users to create and share knowledge with ease when browsing through web content. The usability and ease of use of the web annotation system is evaluated through controlled experiments in lab setting, with the help of participants.
- C4** The design and development of a first of its kind test bench that evaluates the robustness of textual anchoring algorithms. The proposed test bench utilizes real-world web documents that have been manually curated to simulate various forms of web page decay. The test bench is critical as it not only enables the evaluation of the proposed anchoring algorithm (C2), but also provides a benchmark for evaluation of anchoring algorithms that would be proposed in the future.

1.5 Dissertation Outline

In line with the above mentioned RQs and contributions, the dissertation is organized into eight chapters.

Chapter 1 provides an overview of the challenges and issues being encountered on the web. The identified problem statements, corresponding RQs, methodologies and contributions made in this dissertation are laid out.

Chapter 2 provides an overview of related work on web annotation systems and knowledge management. Then, the chapter describes recent studies on the web page decay challenge and establishes arguments on why it is relevant to move away from conventional keyword anchoring algorithms. The chapter also delves into system features that are essential for web annotation systems, especially when they are used as tools for knowledge co-creation.

Chapter 3 explores standard models and framework currently being utilized in knowledge management life-cycles. It then presents a novel framework for web annotation activities that supports creation and sharing of knowledge through web annotations.

Chapter 4 provides an overview of the state-of-the-art anchoring algorithms and then describes our novel *DOM-oriented edit distance* approach that enables stable reattachment of textual annotations. It also describes, how the robustness of the proposed algorithm is evaluated through controlled experiments, and elaborates on the requirements for further evaluations.

Chapter 5 details the web annotation system that is built around the novel anchoring algorithm. The chapter describes the processes and technologies utilized in the development of the artifact. Then, the system features of the developed artifact are explained, together with how these features support knowledge co-creation.

Chapter 6 describes the proposed anchoring algorithm test-bench, and details how the test bench is developed as benchmark for future anchoring algorithms.

Chapter 7 presents additional findings of this work that are not directly related to the RQs of this dissertation, but are still essential as they can potentially help in improving future web annotation systems (and also other crowd-oriented platforms). The chapter explains the limitations of the different evaluation methodologies used in this work, summarizes the dissertation, and highlights its scientific contributions. Finally, the chapter concludes by providing an outlook for future research.

2 Background and Related Work

This chapter discusses the background and related work of our research efforts within the scope of the dissertation. Here we delve into specifics on how web annotation systems have evolved over the years, where they stand today, and how the issue of web page decay is affecting web annotation systems. We also describe how web annotation systems are currently being used for knowledge sharing, and establish what new system features are required to improve users' knowledge sharing experience.

2.1 Annotations

As we alluded to in Chapter 1, annotations are additional pieces of information, typically explanations and comments, that are added to specific parts of media. Conventionally, annotations are added to textual materials, primarily books, and can include several activities such as highlighting texts (by underlining or marking text using hand-drawn circles), drawing scribbles or doodles around the margins, adding comments or explanations and more.

The primary purpose of an annotation is to add additional context to the underlying text that has been annotated, or to highlight parts of text that are useful or interesting for the reader. Annotations are typically made by the readers themselves, or some third person, in most cases not the author of the work. From a cognitive perspective, annotations have always played a critical role in learning and instruction. Highlights, labels, comments and explanations that are added to textual contents, act as a visual aid for the readers of the text, and help them focus on the specific parts that have been annotated. Furthermore, they act as typological representations that link together categories defined by the readers to those defined by the author.

In part, the advent of the Internet, and the plethora of technologies that have followed, today allow web users to annotate content beyond conventional text. Today, annotations are not only used to highlight digitized texts, but also images, videos, and other more complex forms of digital media such as 3D models. This evolution in the usage of annotations has allowed several novel forms of annotations to emerge, for instance: corpus annotations [44], semantic annotations [172, 194, 32], medical annotations [124] and genome annotations [171]. This dissertation focuses on textual annotations on web pages as documents. Textual annotations are an extremely widely used form of annotations. Textual annotations are perceived as useful and particularly easy to use, which makes them easy to adopt. Therefore, we predict that textual annotations will stay with us also in the future as an important tool and even increasingly important tool.

Textual annotations on web documents can be of two types. First, a simple highlight, where the reader selects some content in a web document and would like to store it for later use, such that, whenever the user revisits the web document at a later point in time, they are able to find the previously highlighted content without the need for reading through the whole web page. The second type of textual annotation is where the reader first highlights some content in the web document, and then adds a comment (mostly in textual form) to that highlight. The goal of such annotations is typically to provide some context to the annotated text. In some cases the reader may also want to share the highlighted texts and their corresponding comments within a group or community. Both of these annotations types are intrinsic to web annotation activities today, and are therefore available in all web annotation systems by default.

2.2 Web Annotation Systems

Web annotation systems are ICT tools that enable web users to highlight, comment, and share textual annotations on web pages. The idea of enabling annotations on the web, was first introduced by the National Center for Supercomputing Applications (NCSA) through the Mosaic browser in 1993 [19, 16]. Unfortunately, the feature supporting annotations in Mosaic 1.1 had several “caveats” or issues [23]. First, annotations created on Mosaic could only be shared among a small number of users working in local groups. And second, there was “no security in place”, and thus any user within the group could delete the annotations created by any other member of the group [31]. The technologies that were used in the development of Mosaic were soon adopted by private organizations, and led to the development of the Netscape browser. And in 1997, the NCSA finally discontinued the support for Mosaic.

The web annotation feature of Mosaic, however, was not adopted by the new emerging web browsers (of the time) right away. That said, between 1993 and 1997, few novel web annotation systems were indeed developed, unfortunately, most of these systems were soon discontinued due to lack community support [18]. Some of these earliest (browser/server dependent) annotation systems included CoNote [57], HyperWave [122] (proposed by one of the creators of the Hyper-G Network Information System [73], Hermann Maurer in 1996) and Multivalent Documents (MVD) [155]. The earliest (platform-independent) web annotation system available for general web users, called CritLink [18]; was first proposed by Yee in 1998 [18], and then again in 2002 [188, 17, 75]. Building on Ted Nelson’s concept of Hypertext [127, 128, 14] and ZigZag [130, 15], CritLink introduced four novel hyperlinking features (for web annotation systems), namely *bi-directional links*, *extrinsic links*, *typed links*, and *fine-grained links* [188]. The bi-directional links allowed users to create links between documents that could be followed both ways, i.e., users could attach an originating anchor to a target anchor (as already enabled by the HTML `<a>` tag), but also vice-versa. The extrinsic links allowed users to establish links that linked multiple documents together, even if the users were not the owners of these documents; Yee claimed that this feature would support better collaboration. Through the typed links, CritLink allowed its users to establish a relationship between two anchors. Although this feature (i.e., typed links) was inherently available in HTML, Yee argued that, in practice, it was “universally ignored” [188]. Finally, the fine-grained links enabled CritLink’s users to create links between parts of documents, contrary to conventional links that only allowed linking between documents.

Between 2000-2010, many other web annotations solutions like CritLink (such as Annoty, e-Marked, Gibeo, Third Voice, and YAWAS) were proposed [75], however, all of these projects were discontinued only within a few years (as early web annotation systems were not so popular among general web users). One such solution, developed by the W3C, was the Annotea project, that aimed to establish standards for web annotations as part of the semantic web [12]. Although the project was discontinued, the underlying framework for web annotations first proposed in the project, was later reintroduced [163, 162] in a more stable form as the Web Annotation Data Model in 2017 [13].

2.2.1 Web Annotation Systems before 2010

Among the several web annotation systems launched between 2000 and 2010, those that gained most support and are currently still used are: Diigo [11], A.nnotate [2] and Genius [4]. Unfortunately, all of these web annotation systems are proprietary, and thus, details of their functionalities and features have not been explored in scholarly literature.

Diigo This web annotation system was launched in 2006, and was meant to serve as “so-

cial bookmarking website” [11]. The platform allows its users to bookmark and tag web pages through use of highlights and sticky notes; users however are required to log in to the platform if they want to use it. The platform can be accessed through various operating-system- and device-specific apps and extensions. Unfortunately, the specifics of the anchoring algorithm used by the system are not disclosed. However, because the platform allows users to perform full-text search on cached web pages, it can be assumed that the platform uses some form of *diff-match-patch* algorithm [5] (current state-of-the-art) to search and reattach anchors.

A.nnotate This is a service for annotating and storing web documents. Launched in 2008, this proprietary tool allows its users to add annotations to different forms of web media [2]. Users can either upload their documents to the platform (as PDFs or in Office formats) or, can directly annotate specific web pages using URIs. The documents are then rendered as images, over which users can add marking, comments and more. A.nnotate’s approach to web annotations is distinct from other web annotation systems as it allows reattachment of annotations on both textual materials and images (i.e., text rendered as images). Additionally, the platform allows its users to collaborate through *threaded discussions*, wherein users can add replies to other users’ comments and highlights.

Genius As a web annotation system, this platform allows its users to add annotations to song lyrics, news stories and web documents [4]. Similar to Diigo and A.nnotate, Genius is a proprietary platform, too, and hence, details of the platform’s anchoring algorithm are not available in the literature. Unlike the previous two platforms, Genius is more oriented towards users who are interested in music and poetry, and was originally designed as crowdsourced site with a focus on hip-hop. Users of the platform are allowed to upload, modify and archive lyrics, bios and other details for various forms of music, which is then made available to other users on the platform. The platform also comes with a separate web annotator that can either be integrated by web page owners into their websites, or can be used by web users by adding a prefix (*genius.it/*) in front of any URI. This annotation service, however, has recently been discontinued and is therefore not available for use any more.

A common theme with all of the above mentioned web annotation systems is that all them are proprietary, and therefore, the annotations made on these platforms are not accessible on the open web. Users of these platforms are required to log in to these services if they want to access the platforms or their annotations. Furthermore, the technologies and algorithms that are used by these platforms are undisclosed, making it more difficult for researchers to examine these systems or to even use them for experiments with real-world users.

2.2.2 Hypothes.is

Launched in 2011, the Hypothes.is web annotation system is an open-source project (available under the BSD 2-Clause “Simplified” license) that allows its users to annotate web content on the fly [7]. The platform’s users are allowed to store their annotations as either public or private posts. This entails that public annotations can be viewed by all users of the system without even logging in on to the platform. Similar to Genius, users of Hypothes.is can access the annotation system by browsing to the Hypothes.is website or by simply adding the prefix *via.hypothes.is/* to any web page’s URI. By doing so, users can view all public annotations already made on the web page. Another novel feature of the system is that it employs a *reputation system* for rating user comments.

As a consequence of the openness of the Hypothes.is, the platform has garnered tremendous support from both web users and research communities over the years. This is reflected by the number of annotations that have been created by the platform’s users since its launch. In 2019, the platform achieved its 5 millionth annotation, in 2020 its 10 millionth, and in 2021 the platform already hosts more than 20 million textual annotations [9]. Additionally, given that the underlying algorithms and codes used in the platform are open-source, several scholarly articles have explored the platform [43, 152], while some researchers have also used the platform for conducting controlled experiments [101, 120]. For example, Kalir et al. explored the conventional and disruptive features of web annotations using Hypothes.is, wherein they wrote a scholarly article [99] in collaboration with publicly invited web users and experts using the open platform.

Prior to 2017, the Hypothes.is platform used the Annotator project’s [3] anchoring strategy. This anchoring approach relied on *XPaths* of HTML *DOM* elements (which we describe in Section 4) to detect and reattach anchors for annotated texts. In 2017, Hypothes.is proposed and adopted a novel *fuzzy anchoring* approach to enables reattachment of anchors using keyword matching; the new approach combines Brush et al.’s work [45] with Google’s *diff-path-match* algorithm [5]. However, as stated previously, the new approach is unable to tackle the transient nature of today’s web content; and as a consequence, researchers have found that many of the annotations created on Hypothes.is are already orphaned or are at risk of being orphaned soon [26].

2.2.3 Web Annotation Data Model

An alternative approach to web annotation systems proposed by the W3C, is the Web Annotation Data (WAD) model [13]. Unlike conventional web annotation systems, where developers build and host web annotation tools as platforms on the web; the WAD model combines a standard description model, format and protocol that developers are meant to integrate into their websites and services, whenever they want to enable their website’s (or service’s) users to annotate the website’s content. The building blocks and the vision for the data model were first proposed as part of the W3C’s Annotea project [12], and later built upon by Sanderson et al., who proposed it as the W3C Open Annotation Data (OAD) model [163, 162]. Finally in 2017, the OAD model was finalized into the WAD model [13].

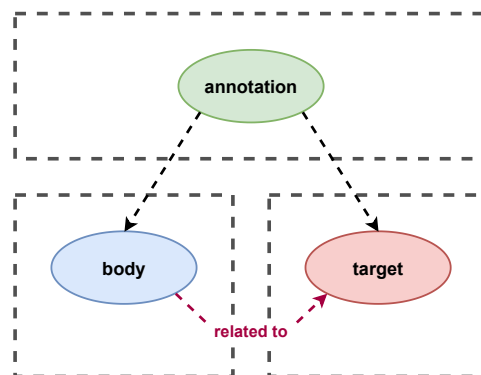


Figure 1 - Illustration of the W3C’s WAD model. adapted from [13]

The proposed framework provides specifications for the *data model*, *vocabulary* and *protocol* in a structured format through which web annotations can be shared and reused across different hardware and software platforms. The data model describes an annota-

tion as having three components: *a body*, *a target* and *a relation* (as illustrated in Fig. 1). The proposed framework utilizes a wide range of selectors (namely fragment selector, CSS selector, XPath selector, text quote selector, text position selector, data position selector, SVG selector, and range selector) to identify content that an annotation is added to. The framework and its corresponding recommendations are designed such that web annotations can be treated as web resources, and therefore are interoperable across devices and platforms.

Unfortunately, the WAD model has yet to be widely adopted by developers and web service providers. Publishers, technology firms, scholarly websites, and academic institutions (such as arXiv, CrossRef, HighWire, PLOS, Project Jupyter, Wiley, and numerous others [10]) have instead opted to integrate the Hypothes.is web annotation system into their websites and web services [152].

2.3 Annotations as a Means for Knowledge Co-Creation

Knowledge co-creation is “a synergetic process of combining content and process from disciplinary traditions to synthesize new ways of knowing” [161]. It is understood as a collaborative process that involves knowledge sharing, aggregation and creation [95, 89, 150, 105]. Analogous to knowledge integration, co-creation is a continuous process through which individuals acquire knowledge and then assign meaning to it, through social practices [158]. As described by Regeer et al. [158], knowledge, communication and behaviour are inseparable. Historically, knowledge was assumed to be something that was held by individuals (in particular, any kind of experts), who imparted this knowledge as “unequivocal and factual” black-boxes to general public. However, over time it was realized that personal views can also add to knowledge, and, consequently, perspectives and opinions started playing a critical role in research and problem solving [158]. That said, knowledge at an individual’s level can therefore be understood as an amalgamation of information available to an individual combined with their personal beliefs, views and insights. It is important to note here, that information relies on the reputation or authority of its “creators” while knowledge is dependent on the audience receiving the information [158].

In context of ICT and the web, information is the content that is available for web users’ consumption on the web (provided by website/service owners), whereas knowledge is interpretation, explanation or association web users draw from the provided information. With this distinction in mind, over the years researchers have proposed several methodologies for computer mediated/supported knowledge sharing and aggregation [160]; and one such class of tools are web annotation systems. Conventionally, the very objective behind creating an annotation was to store it for future use. However, the advent of web has enabled web users to not only create and store annotations, but also share them. Consequently, “annotation sharing” and “annotation communication” were used analogously to “knowledge aggregation” [160]. Although several attempts were made to encourage personal and cooperative annotations [58], as can be seen from the several web annotation systems that were developed prior to 2010, the use annotations for knowledge sharing was first proposed by Robert in 2009 [160]. The author [160] argued that, knowledge aggregation requires users to gather information and knowledge from divergent sources, and therefore the process of using web annotations for knowledge sharing could be viewed as follows [160]:

- The information collaboration process should involve at least two participants.
- There should be at least one cycle of annotation communication (i.e., exchange of annotations) between the participants.

- Annotation should be created in a collaborative yet distributed manner.
- Several annotations should be made over a specified period.
- All annotations should be shared electronically.
- Annotations should be based on each other (i.e., they should be interlinked or connected.)

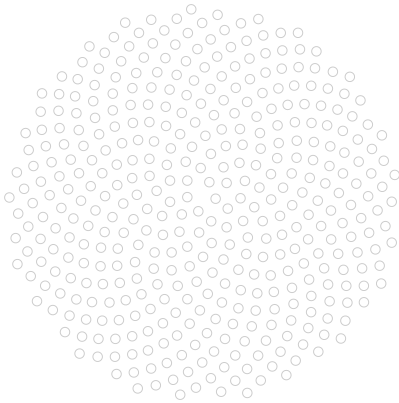
It is important to take note of these processes, as they are useful in not just development of collaborative web annotation systems, but can also be used to evaluate the usefulness and ease-of-use of knowledge sharing ICT solutions (including web annotation systems). We use these processes as guidelines in the development of the proposed artifact in Chapter 5.

Although several scholarly articles have explored the use of web annotations for knowledge sharing and learning in distinct environments such as organizational settings, academics, and healthcare among others [71, 187, 27, 25, 97, 98, 51, 193]. However, the underlying processes for knowledge sharing presented in these scholarly works have remained similar. Even though knowledge sharing processes can be viewed at different levels of abstractions such as the ones suggested by Bagayogo et al., i.e., “initiation, transition, and normalization” [27], these processes always involve actions for sharing, aggregating and creating knowledge and information. Also, in almost all cases the proposed processes or utilized web annotation systems are evaluated and studied in lab settings with participants carrying out predefined actions over a specified period of time. After the experiments, the participants are asked to fill survey forms, that are then used by researchers to gauge the participants’ feedback on the usefulness and usability of the proposed systems and frameworks.

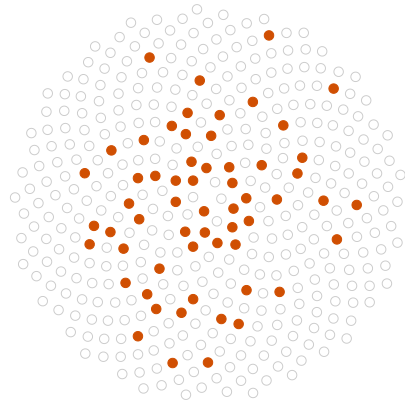
In Chapter 5, we employ the same strategy for evaluating the artifact proposed in this dissertation. Also, drawing influence from the studied literature, we are able to identify key system features that must be included in a web annotation system if the system is to be used for knowledge exchange and learning. These features include support for highlights, comments, questions and sharing, as well as methods for organizing annotations. As pointed out by Kalboussi et al. [97], many of the system features required to support knowledge exchange and learning (on the web) are already available in other online communication environments like: discussion forums, blogs, and wikis. And thus further research is required to identify specific learning tasks that can be supported by current web annotation systems [97]. These specific learning tasks however are yet to be explored.

2.4 Link Rot and Web Page Decay

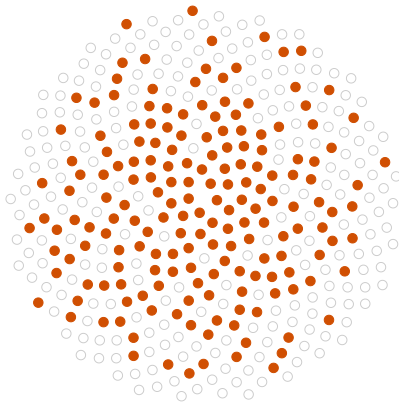
The phenomena of link rot and web page decay is a challenge that researchers of ICT have been trying to tackle ever since the inception of the web. As stated by Schneider et al. [167], the web and its content are ephemeral and transience by nature. This is to say, that content on the web tends to change over time, and as have been observed in literature this behaviour of web content is getting worse as more data is being produced and shared on the web. Although several attempts have been made to preserve content on the web (the oldest and most discussed in literature being the Internet Archive’s Wayback Machine developed in 1996) the issue still remains [153]. Most archival solutions that have developed so far are designed to archive complete web pages or websites (as web documents), and are based on the premise that contents on websites vanish whenever websites change or are shut down.



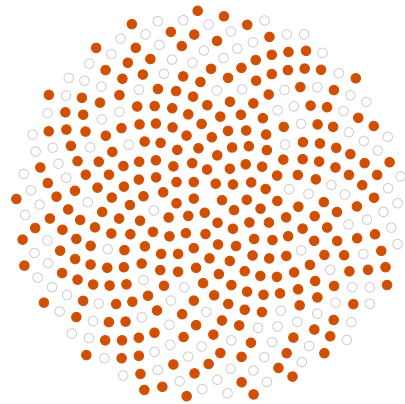
Links start healthy. New citations have been vetted and verified.



After one year. After a year, over 20% of cited links may be dead or otherwise inaccessible.



After five years. After five years, the situation is much worse — over 50% of cited links can be affected.



As time goes on. Link rot is inevitable and rarely reversible. The longer the wait, the more likely a link will have rotted.

Figure 2 – Illustration of decay of cited links with respect to time. adapted from [6].

Since the early days of the WWW (early 1990s) the issue of web decay has only deteriorated further. Websites today often change their user interfaces (UIs), designs and underlying technologies every few months, while contents on most web pages change every few days (some even every few hours)(see Figure 2). Also, most web pages are regularly updated and are often filled using content queried from dynamically evolving databases. Furthermore, there is the issue of link rot, where citations and URIs linking contents from different sources can change over time. These link rot can not only affect the usability of search engines, that direct web users to appropriate web pages based on specified keywords, but can also have severe consequences on research and academia, as cited URIs in scholarly articles and books can redirect the document’s readers to incorrect texts (or no texts at all). And thus, these rotten references can have severe consequences

on science and research, such as reproducibility crisis.

Considering these issues, researchers have conducted several long- and short-term studies to gauge the veracity of the decay issue. For example, in 2008, Lewandowski examined 70 web pages (40 updated daily and 30 irregularly) over a “time span of six weeks in the years 2005, 2006 and 2007” [115]. Lewandowski found that the indexing patterns used by search engines were often irregular, and this meant that, if users were searching for specific and recent content, they could have ended up being redirected to older versions of the appropriate web pages. Similarly, Oguz et al. [135], revisited an original data set of 360 Uniform Resource Locators (URLs) that they accumulated 20 years ago, and found that only 1.6% of the original samples (6 URLs) were still accessible in 2013, and only two were still accessible in 2015 [135].

In another, much larger experiment, Klein et al. [102] analysed link rot and content drift (together referred to as, ‘reference rot’) [153] in over 3.5 million scholarly articles (in science, technology, and medicine). The team accumulated articles published between 1997 and 2012, and determined whether the HTTP URLs cited in the articles were still responsive. The researchers found that, among the studied articles, one of every five article suffered from reference rot; i.e., the content cited in the articles was unreachable. The results were worse in case of articles that only cited web resources. The authors found that 7 out of 10 web resource citations had rotted [102]. The study also found that out of the studied articles, at least 30% of the most recent studies suffered from reference rot, and among these only less than 25% were healthy, whereas the remaining 75% had at least one rotted reference. The authors further illustrated how using services such as the Internet Archive, Memento and Perma.cc can enable researchers in preventing such reference rots.

Similarly, Kumar et al. found in a study from 2015 [108], that among the 406 studied scholarly articles published in two Emerald journals (between 2008 and 2012), more than 2400 references out of approx. 10,000 references (i.e., 23.81%) were URLs. Among these reference, approx. 49% references were already rotten. In total, the researchers found that 39% of the references were redirecting to a HTTP 500 ‘page not found’ error message [108]. As examined by Krol et al. [106], the phenomena of link rot and decay are well researched in literature, and several solutions (not only through archival [195], but also prediction [191]) to tackle both issues already exist, however, in the current digital ecosystem, “it is not possible to completely eliminate the [...] phenomena” [106].

Given this detailed understanding of the link rot and web page decay phenomena, in this work, we adhere to the following formal definitions of these issues as provided in literature:

Link rot: When web resources used to identify URIs cease to exist and therefore no longer provide access to referenced contents, it is referred to as link rot [102].

Content drift: When web resources used to identify URIs change over time such that they cease to represent of the contents that they were originally referencing to, while the originally referenced contents are still accessible (but at a different URI), the phenomenon is referred to as content drift [102].

Web page decay: This phenomenon can be understood as combination of changing content on web pages combined with the issue of link rot, content drift or both (i.e., reference rot). Change in web page content is key when measuring decay as doing so purely on the basis of dead links can be “naive” [30].

A rather unconventional approach to tackle the issue of preserving the web could be to use a multilayered approach. As suggested by Anderson [22], preservation of web content

through the use of web 2.0 could be impacted by six concepts, “individual production, harness the power of the crowd, data on an epic scale, architecture of participation, network effects, and openness” [22]. These concepts are explained by Anderson [22] as:

1. Large volumes of data being produced on the Internet today are stored in privately owned data silos of organizations such as Microsoft and Google. As argued by Anderson [22], the volumes of data and information stored in silos could be considered as a kind of archive, however, this means that the content/service providers can decide to change or remove significant parts of web content whenever they want [22]. Hence, if web content is to be preserved, it would require solutions that do not completely rely on content- and service-providers.
2. Archival solutions can obtain and collect volumes of underlying data from web documents, however, they are seldom able to reproduce the ‘intelligence’ of the services where the data is captured from [22]. Also, archiving is a tedious process and relies on its developers to decide which web pages and sites to archive. Although open web archiving solutions allow users to archive any and all web pages they find relevant, the process of archiving individual pages is often manual and therefore slow. A faster alternative to this is to harness the power of the crowd and crowd-source the task of identifying relevant web pages (as has been illustrated in [76, 69, 68, 47]).
3. Because large volumes of web pages need to be captured and stored in a short period of time, the process of archiving needs to be fast and automatic. Also, it requires considerable amount of processing capability and storage capacity [22].
4. The provided solutions need to be ‘cool’ [22], in order to draw more users to the system [22]. This is critical because, as more people contribute to system, the better we will be able to preserve the web; however, this also means that the system must have good-looking graphics and should be easy to use (unlike conventional repository systems).
5. Solutions should use the power of *network effect* [116, 103]; that is to say, that the content stored on archival solutions should be link-able, as content without connectivity has less meaning [22].
6. Tools and services available for web preservation can not be required to not withhold user data, as this would conflict with the idea of storing data in epic volumes. At the same time, users should be allowed to move or take back their data at will [22]. This has been enabled to some extent, for example, through the EU General Data Protection Regulation (GDPR) [182], however, the issue still remains a complex topic [179].

As explained by Anderson [22], the above mentioned ideas for web preservation can be enabled by web services such as: blogs, wikis, media sharing platforms (like YouTube), data mash-up services (like Google Map APIs), podcasts, social tagging, and social networks [22]; we however argue that none of the state-of-the-art web annotation systems or archival solutions currently supports all six of Anderson’s ideas [22], and therefore in this dissertation, we fill this gap through a novel web annotation system that inculcates these ideas as system features. We build on these concepts in Section 5, where we describe the design choices and thinking behind the proposed web annotation system, Tippianee.

2.5 Losing Annotations

The issues of web page decay and link rot can be observed beyond the conventional discussions on loss of web content and broken citation and reference links. As indicated by recent studies, these issues have now started to affect other forms of web resources, such as DNS records [96, 156] and web annotations. Literature refers to this phenomena as orphaning. First proposed in context of DNS records, the term ‘orphaning’ refers to a DNS server which has an address record within the DNS, however, the domain to which the server belongs is no longer available in DNS records [96]. Similarly, in context of web annotations, an annotation is said to be orphaned, when the content that an annotation is attached to is changed or removed, and therefore it is not possible for a web annotation system’s anchoring algorithm to reattach the annotation at the correct location (if at all) [26]. Here, anchoring refers to the process of searching through a web document and identifying the correct location where an annotation (as highlight or comment) was made.

Because this dissertation explores the use web annotations as a means for knowledge exchange, we focus on the issue of orphaning of web annotations; specifically, with respect to the Hypothes.is web annotation system, as the platform is currently the most used web annotation tool on the web. As discovered by Aturban et al. in 2015 [26], many of the annotations created on Hypothes.is are already orphaned and more than half of the annotations are at risk of being orphaned soon. In their study [26], the authors investigated 6281 highlighted text annotations created on Hypothes.is, and found that 27% of these annotations could not be reattached, i.e., the annotations were orphaned. Aturban et al. [26] found that among the orphaned annotations, only 3.5% of the annotations could be reattached using archived versions of the annotated web pages. Also, 61% of the studied annotations were at risk of being orphaned whenever the underlying annotations (or their web pages) changed. It is critical to note here that at time of Aturban et al.’s study, the Hypothes.is platform had already adopted a new fuzzy anchoring approach to detect and reattach annotations [8]. The new approach utilizes information such as XPath’s and strings to identify where an annotation should be reattached (we discuss the approach in detail in Chapter 4).

Considering the impact that orphaning can have on annotations, in particular, when they are created, used and shared in support of knowledge co-creation, we find it critical to develop a stable anchoring algorithm that can robustly deal with ephemeral web content. To this end, in the next chapters of the dissertation, we discuss the various solutions we developed to tackle the above mentioned challenges.

3 Web Annotations as a Tool in Support of the Knowledge Management Lifecycle

This chapter addresses the research question, RQ1:

“How to support knowledge creation and sharing through web annotations over the web?”

In this Chapter, we discuss how web annotation systems can support knowledge management in groups and communities. We first explore traditional models that are conventionally used to enable knowledge co-creation, and then delve into the SECI model for knowledge creation [132, 133]. We explain a novel annotation framework we created based on the SECI model, and discuss implications of the framework on knowledge co-creation both in organizational settings and on the web; in doing so we address the research question RQ1, by answering RQ1.1, RQ1.2, and RQ1.3.

This research was originally presented in **publication I** and corresponds to the Dissertation Contribution C1.

3.1 Overview of Frameworks for Knowledge Transformation

A prominent theory on knowledge co-creation is the theory of knowledge transformation proposed by Carlile in 2002 [49, 50]. In the theory, the author describes knowledge co-creation as an object-oriented process that revolves around “boundary objects” that can have multiple meanings based on different contexts the objects are viewed with respect to [49, 50]. Carlile argues that it is these boundary objects that allow collaborating groups (made up of individuals with varying backgrounds) to translate one group’s knowledge into context that is understandable by another. And it is this process of translation that enables creation of new knowledge. The author refers to these community-centric contextual boundaries as “knowledge boundaries” [49], and describes them as “both a source of and a barrier to innovation” [49]. According to Carlile, transformation of knowledge from one context to another is what leads to innovation, and as a consequence of this transformation, shared knowledge gains new meaning (different from its source context). This transformation process however, can often be costly both with respect to time and resources, and therefore can also be a hindrance to innovation [50]. In context of organizations, knowledge transformation is a continuous cycle of knowledge retrieval, transformation and storage; and can be described as having three levels of communication: syntactic, semantic and pragmatic [50]. The syntactic level is the level where individuals share a common vocabulary, and therefore can understand each others signals and symbols. The semantic level is based on the premise that even with a common syntax, different signals can be interpreted in different ways (by different individuals), and therefore it suggests that hidden assumptions of different groups that exchange knowledge should be made explicit. The pragmatic level of communication, on the other hand, focuses on understanding the consequences that the exchanged knowledge can have, for instance, understanding the effects of letting go (or losing) the accumulated knowledge [50].

Another well-discussed viewpoint on knowledge co-creation is Paavola et al.’s theory of triological learning [139]. The authors explain triological learning as a form of learning where individuals gain knowledge by developing and transforming shared objects (or artifacts) through collaborative actions. The theory is based on the concept of monological and dialogical learning. The authors explain monological learning as acquisition of knowledge that is purely based on an individual’s cognitive function as it receives external signals [139]. A simple example of monological learning is the knowledge that an individual

acquires (i.e., learns) from reading a book. Whereas, dialogical learning happens when individuals learn by working or interacting with a knowledge community. The authors state that both learning methodologies can occur simultaneously, and that monological learning can be described as learning of explicit propositional knowledge. While on the other hand, dialogical learning can be described as learning skills that involve transmission and retrieval of tacit knowledge [139]. The authors further argue that for trialogical learning, individuals are not required to assimilate or acquire knowledge from a knowledge community, but instead, are required to develop new knowledge themselves. It is important to note here that trialogical learning requires shared trialogical objects that have developed within innovative knowledge communities. Consider for example written documents such as *ReadMe* file in a Git repo, where individuals first externalize their knowledge (i.e., create and upload the code and its *ReadMe* file), and then other members of the community access and improve the Git repo (i.e., shared objects) using their own tacit knowledge (for example by forking the code). In this case the final version of the *ReadMe* file would represent the combined knowledge of the community, while the process through the which the *ReadMe* file evolved could be viewed as knowledge co-creation.

A third theory/model of knowledge building was proposed by Bereiter in 2002 [33]. The proposed model emphasizes the role of idea creation in knowledge co-creation. The author explains that created ideas and concepts can be viewed not only as conceptual artifacts that are the outcome of knowledge creation, but can also act as primary tools for creating knowledge [33]. The author's model states that, when communities participate in knowledge sharing and exchange through the use of shared conceptual artifacts, these mental models can act as tools to create more knowledge. It is important to note that Bereiter's model [33] assumes that new ideas generated by individuals can act as tools, and that these ideas, when brought to communities, can be used as a basis for generating inquiries and discussions [33].

It should be considered here that all of the above-mentioned models revolve around innovative knowledge communities. However, each of the models/theories has been designed with focus on different aspects of knowledge co-creation, namely, (i) levels of communication, transmission and retrieval of tacit knowledge, and (ii) ideas as artifacts for knowledge creation and exchange. Also, these models primarily focus on knowledge co-creation in general and not within specific environment (unlike Nonaka's model [132, 133] that is built around organizational settings).

3.2 Overview of Nonaka's SECI Model

Among the several models for knowledge co-creation examined in literature, Nonaka's model for knowledge-creating companies is by far the most discussed and researched. Considered as a paradigmatic shift in the study of knowledge co-creation, Nonaka et al.'s knowledge spiral (also referred to as the SECI model) constructs a viewpoint on knowledge creation and exchange processes based on experience with leading Japanese companies [132, 133]. Unlike previous models, Nonaka's model considers knowledge communities as an explicit part of the knowledge creation process. Through their knowledge spiral (illustrated in Figure 3), the authors explain that knowledge is created as consequence of four actions: socialization, externalization, combination, and internationalization (SECI), and it is these actions that transform knowledge from one form to another; specifically from tacit to explicit and vice versa [132, 133]. The authors further state that innovation stems from knowledge creation and exploitation, the relationship between them being "socially dynamic"; and that these can only occur simultaneously [131].

In this context, tacit (i.e., implicit) knowledge refers to various forms of internalized

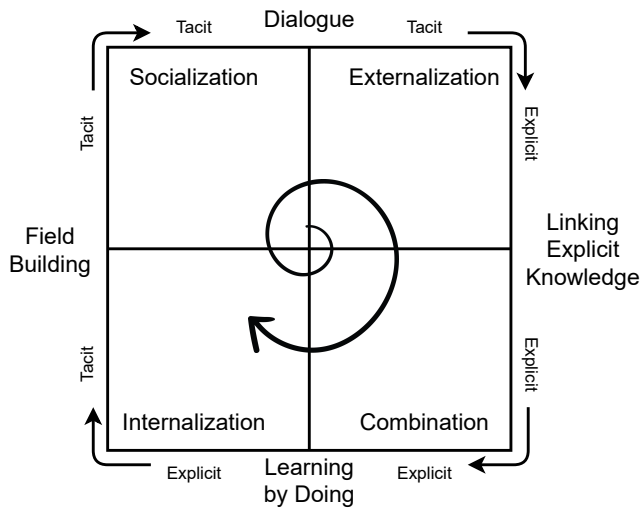


Figure 3 – Illustration of Nonaka et al.'s SECI model. adapted from [132].

knowledge that is difficult to formalize or share. Examples of internalized knowledge are personal experiences, insights and intuitions. Given the personal nature of this kind of knowledge, it is almost always difficult to formalize and share knowledge in this form. Tacit knowledge can also be understood as know-how or embodied knowledge that lies within an individual's mind.

Explicit knowledge refers to knowledge that can be articulated, codified, stored and accessed in some form by the receiver. Common examples of explicit knowledge show in books, guides or manuals. In today's digital landscape, explicit knowledge can be found as web pages (or documents), images, videos and other forms of media. Unfortunately, given that explicit knowledge needs to go through several processes before it is ready to be stored and accessed, the creation of explicit knowledge requires time and resources. Also, as explained by Nonaka et al. [132, 133], it can sometimes be difficult to transform tacit knowledge into an explicit form, as the responsibility of transforming knowledge lies with the knowledge provider. For instance, not everybody has the know-how and knowledge required for writing scholarly articles; one can of-course acquire the knowledge to do so, but may not be interested or might not have the resources to do so, thereby preventing them from being able to convert their tacit knowledge into explicit form. Fortunately, today's web (and social media) has empowered lay users and has made it much easier for web users to share their ideas, experiences, opinions, and beliefs. Consequently, this has led both governing bodies and private organizations to rely on not only the wisdom of individuals working within these organizations, but also the wisdom of web users at-large; particularly, to tackle organizational challenges and solve global wicked issues [183, 100, 111, 175].

One of the most popular example of organizations relying on web users' wisdom (from collective intelligence (CI) research) is the GoldCorp Inc.'s "Goldcorp Challenge". As part of the challenge, the world's leading gold production company (GoldCorp Inc.) posted its proprietary geological data on the Internet, and asked web users to identify potential sites for gold mining, in return for a financial reward. The challenge brought together participants from all walks of life including students, mathematicians, researchers, and (non-expert) web users. Based on the insights (i.e., site locations) the participants came

up with, the organization established its largest mining operation in Canada [186]. Following in GoldCorp's footsteps, several organizations have since then started relying on individuals' implicit knowledge to solve issues that could not be solved by the organizations themselves (for example, OpenIDEO, InnoCentive, among others) [publication V]. This has led to a paradigm shift in organizational problem solving, as more organizations are moving away from traditional hierarchical structures and are involving their (non-managerial level) employees in their business intelligence processes [publication VI].

Going back to Nonaka et al.'s SECI model for knowledge creating companies [132, 133], the authors explain how the transformation between tacit and explicit knowledge occurs:

Socialization This process enables transfer of tacit knowledge from one individual to another. Nonaka et al. [132, 133] describe socialization as a process wherein individuals share knowledge through teaching or mentoring. Here, both the provider of the knowledge and the receiver are in close proximity to each other, and knowledge is shared or acquired through direct interactions. The authors explain that tacit knowledge can also be transferred through common activities such as brainstorming, or even by interacting in common spaces, for example, when getting coffee.

Externalization This process enables conversion of tacit knowledge into explicit knowledge. In this process, the knowledge acquired by individuals or groups is formalized and codified into publishable form. From an organizational perspective, this codified knowledge can be documented that reflects the formal knowledge held by the organization itself, for instance: project reports, recommendations and guidelines.

Combination Through this process, explicit knowledge is passed from one individual to the next. Combination occurs in an organization, when knowledge is merged from several explicit sources. For instance, this can occur when creating new prototypes based on both internal and external knowledge, or can be seen in organizational databases (DBs), where useful information from both inside and outside the organization is captured, cleaned and organized into pre-structured DBs. This new explicit knowledge is then disseminated among the organizations' employees.

Internalization Finally, by internalizing, an individual can convert explicit knowledge into implicit knowledge. Nonaka et al. [132, 133] explain that, as individuals work with an organization, they gain an understanding regarding the organization's inner functioning, they learn from the environment (i.e., learn-by-doing), and in doing so, explicit knowledge of the organization is transformed into individuals' implicit knowledge. This continuous process of internalizing and reflecting enables individuals to recognize patterns, semantics and symbolism that might be well known to other individuals who have worked in the organization for a long time, but may not be known to individuals who have joined the organization recently.

3.3 Web Annotation Activities

Drawing on the studied literature [49, 50, 139, 33], it becomes clear that shared artifacts and objects can play a vital role in knowledge sharing and creation. In context of web annotations, annotations created by users can be viewed as said shared artifacts. This is in line with Robert's proposition (discussed in Chapter 2.3) that web annotations in collaborative environments can be used for knowledge aggregation [160]. However, as stated by Kalboussi et al. [97] if web annotation systems are to emerge as a necessary tool

for learning and knowledge exchange, annotations and annotation activities must evolve beyond traditional highlighting and commenting. As social tools, web annotation systems like Hypothes.is already enable their users to interact over web documents and share their insights. This is distinct from conventional forms of knowledge sharing over the web, for instance, situations where web users share URLs and URLs of web documents with each other and then discuss related topics over social environments like chat or video calls. Doing so requires web users to switch between the sources of shared information and the social environment where the exchange is happening. In case of web annotation systems like Hypothes.is, users can view and comment web document (i.e., communicate) at the same time, without the need for switching between browser tabs (or web pages). This enables users to interact with each other, while having full access to the knowledge (in this case, a web document) being discussed. When viewed in context of the SECI model, this means that socialization is already possible in current state-of-the-art web annotation systems. Also, from an organizational standpoint, different social software integrated or linked to Enterprise Resource Planning (ERP) system too enable socialization beyond the organizations' premises (as illustrated in Fig. 4 by [61, 60]). Such social software can play a critical role in organizational knowledge creation and sense making [81, 184]. A common example of such integration can be seen in services such as Office 365, that enable users (i.e., employees in organizations) to use wikis, chats, and other social features in tandem with organizational systems. It is important, however, to note here that, most web annotation systems being used today are designed as a conversation layer over the web. And thus such web annotation systems can also be used over heterogeneous software ecosystems such as organizational ERPs (as illustrated in **publication I**), that are deployed as websites and web services. That said, by mapping the activities described by Nonaka et al.'s SECI model [132, 133], to annotation activities (both conventional and semantic), we find (in **publication I**) that annotation activities can be extended to knowledge exchange and creation. However, to enable all SECI processes through annotations, it is imperative that annotation systems are designed with two new key features. First, annotation systems should allow users to add semantic descriptions to annotated textual contents (see Sect. 3.3.1), and second, they should enable users to create links between annotations and annotated contents (see Sect. 3.4).

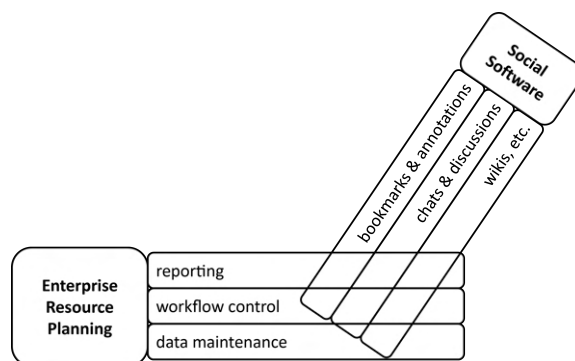


Figure 4 - Illustration of social software features weaved into ERP systems. adapted from [61].

3.3.1 Semantic Annotations

Understood as the process of adding semantic descriptions (as metadata, i.e., data about data) to media, through actions such as tagging, semantic annotations allow web users

to describe different types of media (primarily, textual) in a form that is recognizable, understandable and usable by machines. Through use of metadata stored in ontological forms (i.e., classes, attributes and relations), machines are then able to establish relations between distinct pieces of data and information captured from various sources [40, 82]. These relations can then be presented as knowledge graphs that enable users to visualize relationships (expressed as edges) between objects (expressed as nodes) on the web. Furthermore, these semantically rich web documents and media allow for the creation of linked data, that can enhance users' search experience on the web, and can also support human decision makers in decision making processes [42, 41].

Beyond the conventional machine-centric applications of semantic metadata, semantic annotations are also largely useful in research and learning; as indicated by the plethora of semantic annotation tools developed over the years [137]. In research, semantic annotations allow users to qualitatively analyse texts such as interviews or scholarly articles. By combining semantics and natural language processing (NLP), these tools (such as GATE [56]) allow users to convert (both manually and automatically) pieces of texts into ontological objects (with classes, instances, properties and relations), and thus make it easier for researchers to draw associations between concepts (or keywords). These semantic annotations tools, however, represent a different class of annotation tools (unlike web annotation systems), as their focus is more towards adding semantic metadata to content and not on annotating, storing and sharing content. Furthermore, most semantic annotation tools require some level of understanding of ontological concepts, which are less understood by average web users; and therefore these tools are primarily used in research and academia. Also, almost all semantic annotation tools are designed as closed systems, where if the users want to share semantically annotated texts in groups or communities, they are required to share annotations as separate files (via. social platforms). Conventional web annotation systems, on the other hand, completely lack features for adding semantic metadata to annotated content. We find that it is this gap, that prevents integration of knowledge co-creation processes into annotation systems, and thus, we propose a novel framework that describes how SECI can be enabled through conventional and semantic web annotation activities.

3.4 SECI through Annotation Activities

Based on the above discussion, it can be summarized that, if web annotations (as artifacts) are to be used as a tool for knowledge co-creation, annotation activities must involve actions such as viewing (or reading), communicating, sharing, and aggregation. Keeping these actions in mind, we find that annotation activities (both conventional and semantic) can be described as SECI processes in the following way:

3.4.1 Socialization Activities

As described by Nonaka et al. [132, 133], socialization is the process through which individuals exchange tacit knowledge by means of direct interactions. Whenever two individuals interact with each other, through online chat, in-person discussion or other forms of communication, they apply their views, opinions, experiences and knowledge onto the topic of discussion. As two individuals interact with each other, they apply their own knowledge onto the current state of information (i.e., the original source of discussion and insights from the person who just shared their knowledge), and return new knowledge and information by combining the previously shared pieces of knowledge with their own. As this process of information exchange continues between the two individuals, both individuals gain new insights, which they then include into their tacit knowledge. Given that the

interaction does not lead to creation of some formal knowledge, the exchanged knowledge remains tacit, i.e., it remains in the minds of the two individuals who are part of the discussion.

In context of web annotations, socialization can be understood as the process where system users interact with each other over web annotations or web documents. This entails that web annotation systems should first, allow annotations created by users to be visible to other system users; either as public annotations, visible to all, or private annotations shareable within user groups. By enabling system users to view other users' annotations, annotation systems allow users to establish an annotation as a common point of discussion. Second, web annotation systems must provide users with a means or channel for communication. In the current context, this is enabled through comments, i.e., additional pieces of text added to some annotated content. By allowing users to add comments to other users' annotations, web annotation systems enable users to interact with each other over created annotations. Through this process of to and fro commenting, users are able to share their insights and knowledge with each other. A similar example of this kind of interaction can be seen on discussion forums, which are a well established means of knowledge exchange. Here, it is also important to note that, when exchanging annotations (both simple highlights and comments), users should be able to link multiple annotations to each other. This feature is critical for knowledge co-creation, as a single annotation on a single web document can not be enough to provide relevant context to the topic of discussion. This is similar to a common practice in learning, where one is required to learn from various sources to gauge the whole context (i.e., meaning) of topic at hand. Unfortunately, system features enabling linking of annotations are currently lacking in almost all web annotation systems. At-best state-of-the-art web annotation systems (like Hypothes.is) only allow users to group annotations together through search-able *tags*. We find this practice to be partially helpful, as grouping annotations by adding tags can only provide users with some semblance of structure, whereas allowing users to link multiple annotations together can empower them by enabling creation of (more) structure pieces of information (i.e., annotations). And this can further be useful during the combination phase, as we discuss later.

We argue that, although simple textual comments enabled by conventional web annotation systems support knowledge exchange, they do so only in a limited manner. Consider for example, a scenario where individuals exchanging comments over annotations are not familiar with the underlying vocabulary of the annotated content, or would like to describe the annotated content using their own vocabulary. In both cases, describing annotated content using simple textual descriptions can be time consuming. Also, as more comments are added to the annotation, the context of the annotated content may evolve over time. Here, adding semantic metadata to the annotated content, can make it easier for the individuals discussing the annotation to understand the subject. Also, through the use of semantic descriptions, individuals can add details providing more context to the annotations. We refer to the action of commenting on annotations as *plain annotation* activities, whereas, we refer to the action of adding semantic descriptions as *strict descriptive annotation* activities (illustrated in Figure 5). Here, it is important to note that different individuals can add descriptive annotations based on different (strictly followed i.e., adhering to something that is predefined) ontologies, to the same annotation. When adding descriptive annotations (during the socialization phase), it is not important for different users to be familiar with (or to understand) each other's ontologies, however, given that the descriptive annotations being added revolve around the same annotated text, it can be assumed that users will have some tacit understanding of other users annotations

or ontologies.

3.4.2 Externalization Activities

The process of externalization, i.e., is the conversion of tacit knowledge into explicit knowledge by means of formalization, enables the conversion of an individual's knowledge into a more structured form that is understandable to the explicit knowledge's receivers (i.e., individuals, communities or organizations). An example of externalization in organizations is the creation of project reports and documentations (prepared after meetings or discussions), or following the creation of some artifact that represents the combined knowledge of the individuals who built it. Here, the key aspects are: first, the generated knowledge must be organized in a formal structure, and second, the vocabulary used to describe the annotated content must be known to the knowledge receivers.

In context of web annotations, the process of externalization can be interpreted as some formal knowledge created as a resultant of comments exchanged during socialization activities. One approach to converting comments into a formal structure could be to summarize the comments added to a given annotation. This could be achieved by asking individuals participating in discussions (over annotations) to generate final comments summarizing their discussion by using some form of moderation process. Alternatively, the comments could be summarized using NLP-based solutions, for example, as proposed by Zhang et al. [189]. Here, it is important to note that features and guidelines supporting annotation summaries are currently lacking in state-of-the-art web annotation systems, and in scientific literature, respectively. Drawing from literature [190], we argue that summaries can be critical in enabling creation of formal knowledge/information from comments (added to annotations) and thus should be common practice if annotations are to be used in support of knowledge co-creation.

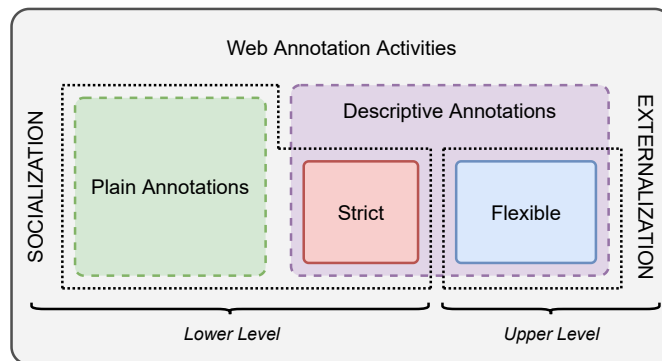


Figure 5 – Illustration of proposed annotation activities and levels corresponding to Socialization and Externalization. Combination (i.e., consolidation) and Internalization (i.e., adoption) here, occur at the user level.

With respect to semantic annotations, externalization can be divided into two distinct tasks. First, suggestion of new vocabularies for semantically describing annotated content. This task is based on the premise, that while individuals are adding semantic descriptions to annotated content, they might encounter the need to propose new classes, attributes or relations. In such as case, we argue that web annotation systems should allow users to suggest new vocabularies (that could later be formalized into new ontologies) on the fly; we refer to this annotation activity as *flexible descriptive annotation* activities (illustrated in Figure 5). From an organizational perspective, this task can be interpreted as

employees proposing ad-hoc vocabularies by utilizing the (tacit) knowledge they gained after the socialization process. The second (semantic annotation) task would be carried out by the knowledge expert of the group (i.e., community or organization). Under this task, the knowledge experts should be enabled to combine the ad-hoc vocabularies proposed by individuals with the already existing vocabularies of the group. They should be further allowed to integrate these vocabularies into new, modified ontologies which can then be adopted by the community or the organization. This integration of vocabularies and ontologies can be understood as formalization of the accumulated knowledge (from a semantic standpoint). It should also be noted that communities and organizations, should be allowed to use multiple distinct ontologies, based on the different topics of discussion.

3.4.3 Combination Activities

The next stage in the SECI model of knowledge dimension is combination. This process, as explained by Nonaka et al. [132, 133], entails the conversion of existing explicit knowledge into new explicit knowledge; in particular, by combining the explicit knowledge already available within an organization with new explicit knowledge acquired from other sources (such as through use of crowdsourcing practices [136, 107] and open innovation [185]). In context of organizational knowledge management, the process involves structuring and storing of new knowledge/information into organizational DBs.

In regards to web annotations, the process of combination can be interpreted as the process of consolidating knowledge, stored in form of annotation summaries (from one or multiple sources). As mentioned for socialization activities, linking multiple annotations (irrespective of the source document) can enable users to organize annotated content (and their comments) into structures. The structures resulting from annotation linking are different from those resulting from the (conventional) tagging method, as the latter only allows annotations to be grouped based on user-defined tags. Consider for example, the way books are organized under classifiers such as authors, years, and genres, or notes are organized based on topics, sources and more. While annotation linking enables structuring of annotations into graph structures where annotations can be linked individually to one or more annotations; thereby, allowing for different kinds of cardinalities between annotations, i.e., one-to-one, many-to-many, one-to-many and many-to-one. With these aspects in mind, knowledge aggregation through use of web annotations can be achieved by generating new summaries from the knowledge (i.e., annotation summaries) created during the externalization process. And this aggregation of previously formalized annotation summaries can be interpreted as creation of new explicit knowledge from multiple explicit sources.

Alternatively, the annotations and their summaries generated during the socialization and externalization process, can also be expressed as annotation (knowledge) graphs, where annotations can be expressed as nodes while their linkings or relationships can be expressed as edges. Such graphs can also be interpreted as formalized knowledge, although not in textual format but rather in graphical form. As illustrated in literature [159, 190, 52], this method of expressing knowledge as graphs can be useful, especially as it is easier for users to gain information when viewing things in graphical form, as compared to viewing the same information in textual form. Here, it is important to note that the proposed annotation (knowledge) graphs are different from knowledge graphs, as the former is a collection of annotations expressed as graphs, while the latter is used to store and express semantically described entities in form of graphs [65, 85].

From a semantic perspective, combination would require an additional step, that would have to be carried out by the group's knowledge expert. This task is about consolidating

the several ontologies and ad-hoc vocabularies proposed by individuals during the externalization process into a single aggregated ontology. Since the task requires understanding of ontologies and vocabularies, the task can only be accomplished by a knowledge expert, and not by other (no-expert) users. Also, combining ontologies requires use of specialized tools (i.e., ontology editors) like Protégé [125], and therefore, the task would have to be performed outside the web annotation system.

3.4.4 Internalization Activities

Internalization is the process wherein individuals assimilate the aggregated knowledge (generated after the combination process) into their own tacit knowledge. To simplify, this could be understood as acquisition and adoption to some new explicit knowledge. Given the non-conscious mental nature of the process, this activity primarily occurs in the knowledge receiver's mind. From a web annotation perspective, this could be interpreted as users enhancing their knowledge by reading through the collected annotations and summaries. Similar to the socialization process, internalization too requires that created annotations, their summaries, and corresponding annotation (knowledge) graphs are visible to members of the community or organization, as only by doing so the generated knowledge can be propagated within the group. Once the internalization process is completed, the annotations and comments associated to the topic of discussion should now be a part of the users' knowledge pool. With respect to semantic annotations, the internalization process can be understood as adoption of the new semantic ontologies developed in the combination phase. Just as for plain textual annotations, the generated semantic ontologies (at this stage) must be shared within the group; and the group members should adopt these ontologies, and use them while adding semantic descriptions to future annotations.

To summarize, the several web annotation activities described above, are based on the SECI model of knowledge dimensions [132, 133]. Because the proposed annotation activities are designed by drawing influence from the SECI model which is a 'spiral' sequence (as illustrated in Figure 6), the activities described above also follow a spiral pattern, that is, after internalization activities, the users can move on to socialization activities, and so on. Based on studied literature, we reason that enabling the above-mentioned activities (through system features) in web annotation systems should allow such systems to emerge as indispensable instrument for accessing, creating, refining and dissemination knowledge (i.e., for the whole knowledge management life-cycle) over the web.

3.5 Discussion and Conclusions

In this chapter, we addressed the challenge of using web annotations as means for knowledge co-creation. We delved into four widely adopted frameworks (and models) for knowledge transformation and creation. And building upon these models introduced various (textual and semantic) web annotation activities in support of knowledge creation and sharing.

In regards to the research question RQ1.1: *"What insights from knowledge management can be incorporated into web annotation processes?"*, we concluded based on literature [49, 50, 139, 33, 160], that in a collaborative environment web annotations can be understood as artifacts that can be used for creating and sharing new knowledge (i.e., knowledge co-creation). We also found that by examining web annotations as shared artifacts, concepts from knowledge management and transformation can be mapped to specific annotation activities.

By delving into the SECI model of knowledge dimensions [132, 133], and by building

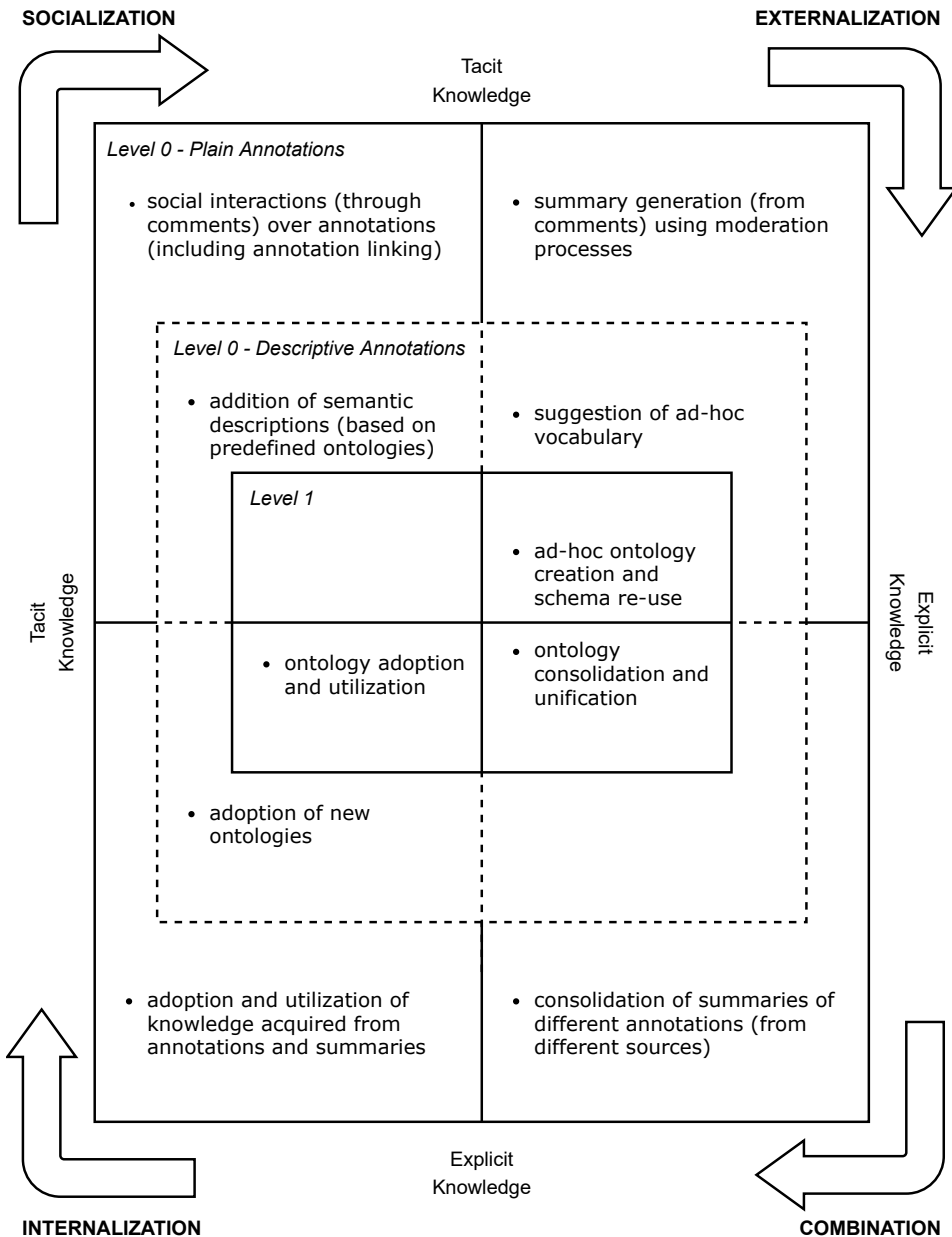


Figure 6 - Illustration of proposed annotation activities (segregated into levels) as interpreted with respect to Nonaka et al.'s SECI model of knowledge dimensions. adapted from [publication 1].

on the notion of using web annotation for knowledge co-creation, we proposed several web annotation activities based on SECI processes (namely, socialization, externalization, combination and internalization). These annotation activities as a whole represent the research contribution that answers the research question RQ1.2: "How can web annotations support knowledge management life cycles?". Based on evidence from literature, we explained how users of web annotation systems could be empowered, to exchange and

create knowledge over web documents. By interpreting textual and semantic annotation types in light of socialization, externalization, combination and internalization processes, distinct annotation activities specific to type of annotations and processes are expressed through various levels (as illustrated Figure 5 and 6).

To refer back to the research question RQ1.3: *“What system features have to be fulfilled by solutions to support knowledge creation and sharing, when using web annotations?”*, by building on Nonaka et al.’s work on knowledge creating companies through the use of SECI processes, we demonstrated, how online communities (through use of web annotations) too could evolve into knowledge creating communities. The identified system features have been described in Chapter 3.4.

Unfortunately, as pointed out, the features required to support the proposed annotation activities are currently not available in the state-of-the-art web annotation tools, and therefore, there exists a technological gap that needs to be filled. We find that, although some of the proposed activities can be enabled through system features (by using current technologies, as we will illustrate in Chapter 5), activities revolving around semantics require further examination. Developing technological solutions enabling the creation of ontologies over web annotation systems requires, first, a better understanding of the complexity of ontology creation processes (both with respect to the process and technology), and second, cognizance of how this understanding could be integrated with latest technologies, such as W3C’s WAD model.

4 A Novel Anchoring Algorithm for Textual Web Annotation

This chapter describes a novel anchoring algorithm, that addresses the issue of orphaning web annotation anchors. First, a brief technical overview of the state-of-the-art in anchoring algorithms is provided, and then the details of the proposed algorithm are described. We then present the experimental setup used for evaluating the proposed algorithm in Chapter 4.2.2. The algorithm was developed as part of Dissertation Contribution C2 and addresses the research question RQ2.1:

“How can web annotations be prevented from being orphaned?”

The novel algorithm presented in this chapter was originally introduced in **publication I**, and later detailed in **publication II**.

4.1 State-of-the-Art

In the context of web development, anchoring refers to creating a link between a source and a destination object. On the web, anchor elements are used to create hyperlinks between a source anchor (such as some text, image, or other HTML element) and a destination anchor (mostly URL to some web page). Here, the keyword ‘anchor’ refers to points between which the link is created. The simplest form of an anchor element in HTML is the `<a>[. . .]` tag, where `<a>` represents an anchor, that typically includes two components, a source object/element and a destination (defined under attribute `href` to which the source object is linked to).

Consider the following for example, an HTML code that is to be linked to some text (lets say, ‘IS’), such that it redirects users to the ‘Information system’ web page on Wikipedia; this in HTML would be encoded as follows:

```
<a href="https://en.wikipedia.org/wiki/Information_system">IS</a>
```

It should be noted that HTML codes like the one just illustrated are typically explicitly typed into HTML documents, this is to say, that the source and destination are known to the documents’ author (or the web pages’ developer). That said, consider the scenario, where the source (i.e., the text ‘IS’) was changed without the knowledge of the author/developer who linked it; or, the destination was changed to a different URL. In both cases the anchor element would not remain the same as it was designed by the original author/developer, or to view it differently, the anchor element would not be linked as was intended.

Similar to web anchors, anchors in web annotation systems also have two components, a source media (i.e., some text or HTML element), and a linked highlight or comment (one added by the user who created the annotation). And thus like web anchors, change in the source media of a web annotation (i.e., annotated text) can render a annotation useless for the person who created it. This is more likely to happen, in case of web annotations, as the authority to modify the source of the annotation is beyond the scope of the user who created the annotation. And therefore, if the developer of a web document modifies the annotated document, there is a high probability that the user-created annotation could be lost (i.e., orphaned), especially if the annotated content is a word or a short string.

To tackle such issues in web development, web researchers (in the early days of the web) proposed the use of XPath [86, 24, 21], i.e., the XML Path Language, allowing for querying nodes in XML (and other markup languages like HTML) documents. Through the use of XPath in HTML, developers are able to encode paths of specific HTML elements into

abbreviated syntax derived from the element's position within Document Object Model (DOM) hierarchy (inside the HTML document). Consider the following HTML code (from Wikipedia) as an example:

Listing 1 - Example of HTML code from Wikipedia homepage.

```
<div id="mp-welcome" >
  Welcome to
  <a href="/wiki/Wikipedia" title="Wikipedia" >
    Wikipedia
  </a>
</div >
```

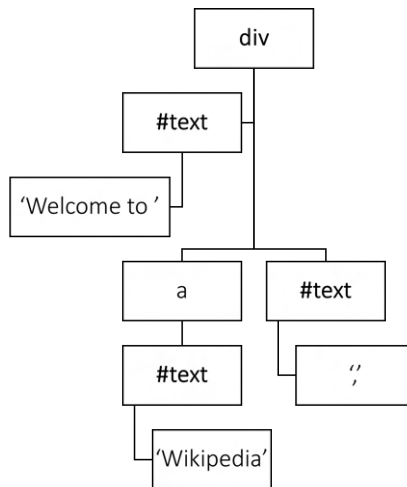


Figure 7 - Illustration of HTML DOM.

The DOM hierarchy of the presented HTML code (see Listing 1) refers to the tree structure of the HTML elements illustrated in Fig. 7. The XPath of the <a> tag in this case can be encoded as follows,

```
/html/body/div[3]/div[3]/div[5]/div[1]/div[1]/div/div[1]/a
```

Early web annotation systems adopted the use of XPath, and thus most annotation systems developed around 2000 utilized anchoring algorithms based on XPath anchoring. In this approach, when a user annotated a piece of text on a web document, the annotation systems stored the annotated text together with the XPath of the HTML element the text belonged to. Using this method, when a user returned to an annotated document, the annotation system searched through the whole document to identify the correct element where the user's annotation was anchored. Once identified, the highlight or comment was reattached to the appropriate location. Alternatively, annotation systems (like the Annotator project [1]) also used additional information such as *string offsets* to identify anchors. These string offsets indicated the position of the first and last annotated characters within their respective HTML elements [8].

Given the simplistic nature of the anchoring approach, any changes to annotated text or the underlying DOM structure rendered the anchoring algorithm ineffective thereby

orphanning the annotation. As web documents became more ephemeral, the orphaning issue became more critical, and thus XPath anchoring were soon replaced with keyword anchoring.

4.1.1 Anchoring Annotations using Keywords

Proposed by Brush et al. in 2001 [45], keyword anchoring is based on the proposition that, when creating annotations, the content being annotated is more significant for retrieval of anchors than the position of the annotated text itself. Brush et al. [45] stated that, if an annotated document changes over time, there are two possible actions that could be taken. First, one could show the annotation as an orphan, or second, one could attempt to reattach the annotation to the modified document. Focusing on reattachment, the authors argued that based on their previous work [46], they found that, when creating annotations, users focused on “key words, proper names and quotations” [46], and therefore, a robust annotation system (if possible) must be able to reattach annotations to the correct keywords even after the web page has changed. Building on this premise, the authors proposed a novel anchoring algorithm that saved textual information around annotated text (see Figure 8).

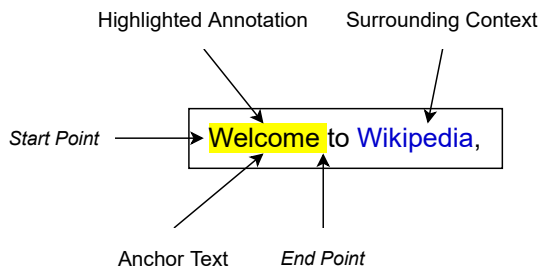


Figure 8 – Illustration of annotated content and context as explained by Brush et al. adapted from [45].

The algorithm focuses on two elements while generating anchors: the anchor text (i.e., the exact text as selected by user) and the surrounding context (i.e., the textual information before and after the selected text). Using these elements, the algorithm stores the following details as the anchor:

1. *HTML bookmark for the selection*: “An Internet Explorer-specific string used to quickly anchor annotations in documents that have not changed” [45]. If the annotated web document is in the same state as it was when the annotation was created, the annotation can be reattached by searching for the (full) annotated string through the document.
2. *Offset from start of document*: Indicating the number of characters between the anchor text and the beginning of the document.
3. *Length of the anchor text*: Indicating the number of characters in the anchor text.
4. *Information start and end points of the anchor text*: Text (i.e., few characters) before and after the the anchor text.

5. *Information about keywords in the anchor text*: This includes list of keywords from the anchor text, together with their offset (both start and end) location. These offsets indicate the position of the first and last characters of the keywords, with respect to the anchor text.

Building on the above mentioned approach, the authors built the algorithm into an annotation systems that ‘plugged into’ the Internet Explorer browser [46]. The annotation system allowed users to visualize textual annotations even when the underlying web document was changed. Additionally, the system presented confidence scores to the users, indicating how confident the algorithm was when reattaching annotations to a changed document.

The approach proposed by Brush et al. [46, 45] is considered to be one of the first attempts at building a robust annotation system. Since then several aspects of their algorithm and annotation system have been adopted by several web annotation systems. In Chapter 5, we discuss how these features can be integrated into novel annotation systems, using today’s technologies.

4.1.2 The Fuzzy Anchoring Algorithm

Prior to 2013, almost all web annotation systems (including Hypothes.is) utilized either the XPath approach, or the string offset approach (used in the Annotator Project [1]), or a combination of both approaches. However, as we discussed earlier, given the ephemerality of the web, it became more and more difficult to reattach annotations to changing web documents. To this end, the Hypothes.is team, organized a crowd-oriented challenge near the end of 2012, and developed a novel anchoring algorithm they called fuzzy anchoring [8]. The term ‘fuzzy’ being an informal reference to ‘approximate string matching’ [8]; one of the four strategies used in the novel algorithm. By combining the state-of-the-art in anchoring algorithms (illustrated in Figure 9), the Hypothes.is team argued that while reattaching annotations the new fuzzy algorithm was able to withstand both changes in web documents’ content and its structure.

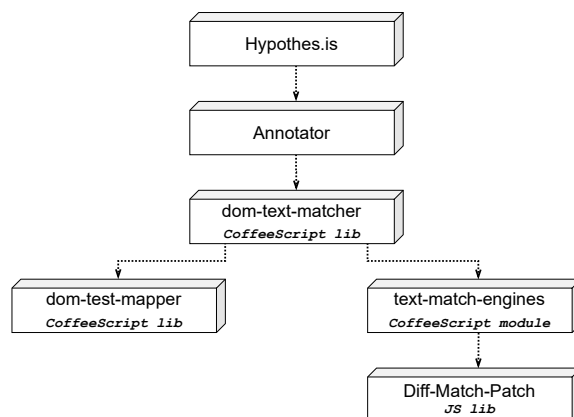


Figure 9 – Architecture of Hypothes.is’ anchoring sub-systems. adapted from [8]

The new approach utilizes three new selectors that store both structural and contextual information of the annotated content:

1. *RangeSelector*: This selector stores the XPath of the annotated DOM elements, together with string offsets of the annotated anchor text.

2. *TextPositionSelector*: In this selector, the start and end offsets of the anchor text (with respect to the whole document) are stored. It should be noted that all offsets are numerical values.
3. *TextQuoteSelector*: This selector stores the anchor text and its surrounding context.
 - (a) *exact*: The exact *TextQuoteSelector* stores the actual anchor text.
 - (b) *prefix*: This selector stores the first 32 characters (immediately) before the anchor text.
 - (c) *suffix*: This selector stores the first 32 characters (immediately) after the anchor text.

Using the above mentioned selectors, the fuzzy anchoring approach utilizes a four-step method to reattach annotations; these are: using the *RangeSelector*, using the *TextPositionSelector*, using context-first fuzzy matching, and using selector-only fuzzy matching [8].

1. *Using the range selector*: This method assumes that the annotated web document has not changed since the annotation was created, and thus searches for the annotation using the information stored in the *RangeSelector*. Once the HTML element is identified within the document's DOM, the algorithm matches uses the information stored inside the *TextQuoteSelector*, and verifies if the anchor text and surrounding context are same, if so, the annotation is reattached to the identified location.
2. *Using the text position selector*: If the first approach is unable to reattach to the annotation, the fuzzy anchoring algorithm moves to the next approach. Here, the algorithm attempts to reattach the annotation using the information stored under the *TextPositionSelector*. This method assumes that the structure of web documents is more likely to change than the content itself.
3. *Using fuzzy matching on context*: The fuzzy matching method (also formally referred to as approximate string matching) used by the fuzzy anchoring approach is based on Google's diff-match-patch algorithm [5]; the latter being based on the edit distance algorithm, first proposed by Levenshtein in 1966 [112]. By measuring the minimum number of primitive operations required to convert one string into another [126], the edit distance algorithm (and thus in-turn Google's diff-match-patch algorithm), allows for numerous applications of string matching and string searching; the most common use-cases being in search engines and in version control systems of platforms like GitHub.

In context of web annotations and Hypothes.is' fuzzy matching method, the edit distance algorithm allows the annotation tool to search through and reattach annotations within large pieces of texts (i.e., web documents). By matching the annotated text with (parts of) the annotated document, and by calculating the minimum edits needed to turn a string into another string, the algorithm is able to identify the most likely position (i.e., piece of string) that was annotated in the first place. Hypothes.is' fuzzy anchoring algorithm also takes into consideration the *TextPositionSelector* when fuzzy matching the prefix- and suffix-*TextQuoteSelector*. Once the algorithm has successfully estimated the prefix and suffix contexts, it checks the exact-*TextQuoteSelector* against the content between the new prefix and suffix contexts. If the *TextQuoteSelector* from the original annotation matches with the current *TextQuoteSelector* values beyond a predefined threshold, the algorithm reattaches the annotation to the identified location.

4. *Using fuzzy matching on selectors*: If all of the above methods are unable to find the anchor text, as a 'last-ditch strategy' [8], the algorithm conducts fuzzy matching through the whole document, and searches for the anchor text in the whole document. It then attaches the anchor to text which it finds textually closest to the anchor text.

Although this approach enables Hypothes.is to reattach most textual anchors, literature from 2015 (only two years after the algorithm was proposed) suggests that 26% of the system's annotations (studied by Aturban et al. [26]) have already been orphaned, while 61% would be orphaned soon [26]. This issue becomes critical, when we examine the tremendous support the platform has gained over the years, especially from academia and research [152].

4.2 The Novel Anchoring Algorithm

Drawing from the issues encountered in the state-of-the-art anchoring algorithms, it can be concluded that, although anchoring algorithms have evolved over the years, the approaches used to find and reattach annotations are still unable to keep up with the ephemerality of today's web content. In our examination of the currently used anchoring algorithms, we found that both the content and the structure of annotated texts are critical, when it comes to reattaching annotations. As suggested by Brush et al. [45], the contents of anchor texts are important to users, and so can be the surrounding context (as illustrated by Hypothes.is [8]). However, none of the state-of-the-art algorithms focuses much on the annotated document structure. Although current approaches consider the location of the anchor texts with respect to the document's DOM, they do so only partially, through use of XPath. Also, as demonstrated by Aturban et al. [26], even when focusing on both contents and contexts, anchoring algorithms still fail in one out of four cases when reattaching annotations. We find that, although this is true, the findings do not show the complete picture. During our examination of the different algorithms, we found (in **publication II**) that of the 3/4th annotations that are successfully reattached, many tend to get reattached at incorrect locations (see Figure 10). This phenomenon can often occur in web documents where same (or similar) content exists multiple times within the same document, for instance, in websites like Wikipedia and Amazon. Also, although web page decay can start within a few days, we find (based on literature) that the probability of content change in web documents is much higher than the probability of structural change, especially over a short duration (i.e., few years). With this in mind, we propose (in **publication II**) a novel anchoring algorithm that focuses on the annotated content, its surrounding context and its structures.

4.2.1 The DOM-Oriented Edit-Distance Algorithm

Similar to the fuzzy anchoring algorithm used in Hypothes.is, the proposed DOM-oriented edit-distance algorithm also utilizes multiple selectors and matching strategies to reattach annotations. The algorithm takes into consideration both the annotated text and the DOM properties of the element the annotated text is part of. This way, the algorithm is able to capture not only the anchor text but also its surrounding context (similar to Brush et al.'s keyword anchoring [45], see Figure 8). Building on fuzzy anchoring, the proposed algorithm also utilizes textual offset values indicating the position of the annotated string with respect to its parent DOM element. Finally, for string matching, the algorithm utilizes Levenshtein's edit-distance algorithm [112] (on which the diff-patch-match algorithm [5] is based on); in particular, the anchoring algorithm makes use of fuzzy string matching.



Figure 10 – Illustration of Hypothesis reattaching annotation at incorrect location. We have that (a) indicates the actual annotated anchor text i.e., “5,710,618”, while (b) indicates the text to which the annotation has been reattached (i.e., “710 articles”). Here, the position shift is caused by fuzzy string matching, as the algorithm finds that “5,710,618” matches more with “710 articles”, and not “5,745,710”. [publication II].

As an additional measure, the algorithm uses a custom-designed DOM matching function that compares DOM attributes and assists in identifying the correct location of annotated text.

The proposed algorithm utilizes two selectors that enable the identification of anchor text, i.e., *#Text Element* and *DOM Node*. The selectors are based on HTML DOM structures and thus, the text element selector stores attributes relating to the textual part of HTML nodes, whereas the DOM node selector only stores non-textual attributes. Here, it is also important to note that, in HTML, all DOM elements have two parts, first, the node itself (example `<div>`) that can have attributes such as `class` and `id`; and second, a child element that can either be another DOM element (such as `<div>` and `<p>`) or a text element, that only stores the text (see Fig. 7).

1. *#Text Element*: This selector stores the annotated text using six attributes:
 - (a) *nodeDepth*: This attribute indicates the depth of the element with respect to the annotated node’s DOM structure, starting from zero (for the root node).
 - (b) *nodeName*: This attribute indicates the name of the DOM element based on the HTML tags (e.g., `<div>`, `<p>`, `<a>`, `<i>`, `<u>`, `<h1>`). In case of the element `#Text`, the *nodeName* is set to “`#text`”.
 - (c) *nodeValue*: This attribute stores the value, i.e., the textual attribute of DOM nodes. The attribute is only used for `#Text` elements.
 - (d) *annotated*: This attribute is a Boolean value that indicates whether the text inside the element is part of the anchor text.
 - (e) *startOffset*: This attribute indicates the start offset of the first character of the anchor text with respect to the `#Text` element. For instance, if the anchor text starts from the third character of the *nodeValue* attribute, the offset is set to two, i.e., one less than the position of the first anchor text character.
 - (f) *endOffset*: Similar to *startOffset*, this attribute stores offset of the anchor text, but this time with respect to the last character of the *nodeValue* attribute.
2. *DOM Node*: Similar to the `#Text` element selector, this selector also utilizes multiple attributes. However, given that the selector only stores information regarding DOM nodes, it is required to store two previously discussed attributes; namely: *nodeDepth* and *nodeName*. And, if available, additional attributes including *id*, *className*, *alt*, *dataset*, *href* and *src* that again correspond to resp. HTML node attributes.

Through use of the above-mentioned attributes and selectors, the proposed algorithm produces anchors as JSON objects, for example:

Listing 2 - Example of JSON object generated by the novel anchoring algorithm.

```
"ww-1540806054738-a2ca6765" : {
  "addedon" : "October 29, 2018 11:40 AM",
  "anchor" [
    {
      "nodeDepth" : 0,
      "nodeName" : "DIV"
    },
    {
      "nodeDepth" : 1,
      "nodeName" : "#text",
      "nodeValue" : "Welcome to ",
      "annotated" : true,
      "startOffset" : 0,
      "endOffset" : 3
    },
    {
      "nodeDepth" : 1,
      "nodeName" : "A",
      "href" : "https://en.wikipedia.org/wiki/Wikipedia",
    },
    {
      "nodeDepth" : 2,
      "nodeName" : "#text",
      "nodeValue" : "Wikipedia",
    },
    {
      "nodeDepth" : 1,
      "nodeName" : "#text",
      "nodeValue" : ", ",
    }
  ],
  "owner" : "abc@abc.com",
  "selectedtext" : "Welcome ",
  "sharedwith" : "LSS",
  "transclusion" : "ww-1540823638993-7270b4f1",
  "urlHost" : "en.wikipedia.org",
  "urlParameter" : "",
  "urlPathname" : "/wiki/Main_page",
  "urlProtocol" : "https:"
}
```

Please note that JSON anchor in Listing 2 corresponds to the annotated content shown in Fig. 8, while the DOM tree structure corresponds to Fig. 7.

While making use of the above mentioned selectors, the algorithm utilizes a multi-tiered approach to detect and reattach annotations (similar to Hypothes.is), as follows:

1. *Fuzzy matching DOM attributes*: Using a custom-designed script for DOM attribute

matching, the algorithm searches through the complete document DOM and iteratively looks for DOM elements that match the root DOM node of the anchor text. At this stage, it is possible that the algorithm finds more than one node that fits this requirement. In such a case, the algorithm matches each of the identified nodes based on their (and their childrens') attributes and generates two sets of matrices; one indicating the probability of matches, and a second indicating the probability of mismatches.

2. *Fuzzy matching text*: While matching the attributes in the DOM matching phase, the algorithm also compares (using fuzzy string-matching) each of the nodeValue values it encounters. The probabilities of matches and mismatches calculated from this phase are again stored into two sets of matrices.

By aggregating the generated match and mismatch matrices from DOM attribute matching and fuzzy text matching, the algorithm generates (based on a predefined ratio) combined probability values. The algorithm then, reads through these probabilities and selects the DOM nodes with maximum matches and minimum mismatches. Finally, the algorithm reattaches the annotation by using fuzzy string matching (a second time) to select the exact annotated text within the identified DOM node. When reattaching annotations, the algorithm returns the final match-mismatch values as a JSON object (as illustrated in Listing 3).

Listing 3 – Example of match and mismatch probability generated by the novel anchoring algorithm, during annotation reattachment.

```
{
  "strIdxMAXmat": 3,
  "strMat": [
    4.98,
    4.98,
    4.98,
    4.98,
    4.98,
  ],
  "strMis": [
    75.89999999999999,
    48.940000000000005,
    9.959999999999999,
    0,
    9.51,
  ],
  "strSimIdx": 0.9960000000000001
}
```

Here, the values under `strMat` and `strMis` indicate the five DOM nodes that the algorithm identified as being similar to the annotated DOM node (see Fig. 8). Each of the numerical values indicate the final probabilities of matches and mismatches, respectively. The object `strIdxMAXmat` indicates the index of the node with maximum match probability and minimum mismatch probability, while `strSimIdx` (i.e., similarity index) indicates the probability (as a float value between 0 and 1) of match between the original anchor text, and the text with the node identified by the proposed algorithm. A detailed illustration of the proposed algorithm is presented in Figure 11.

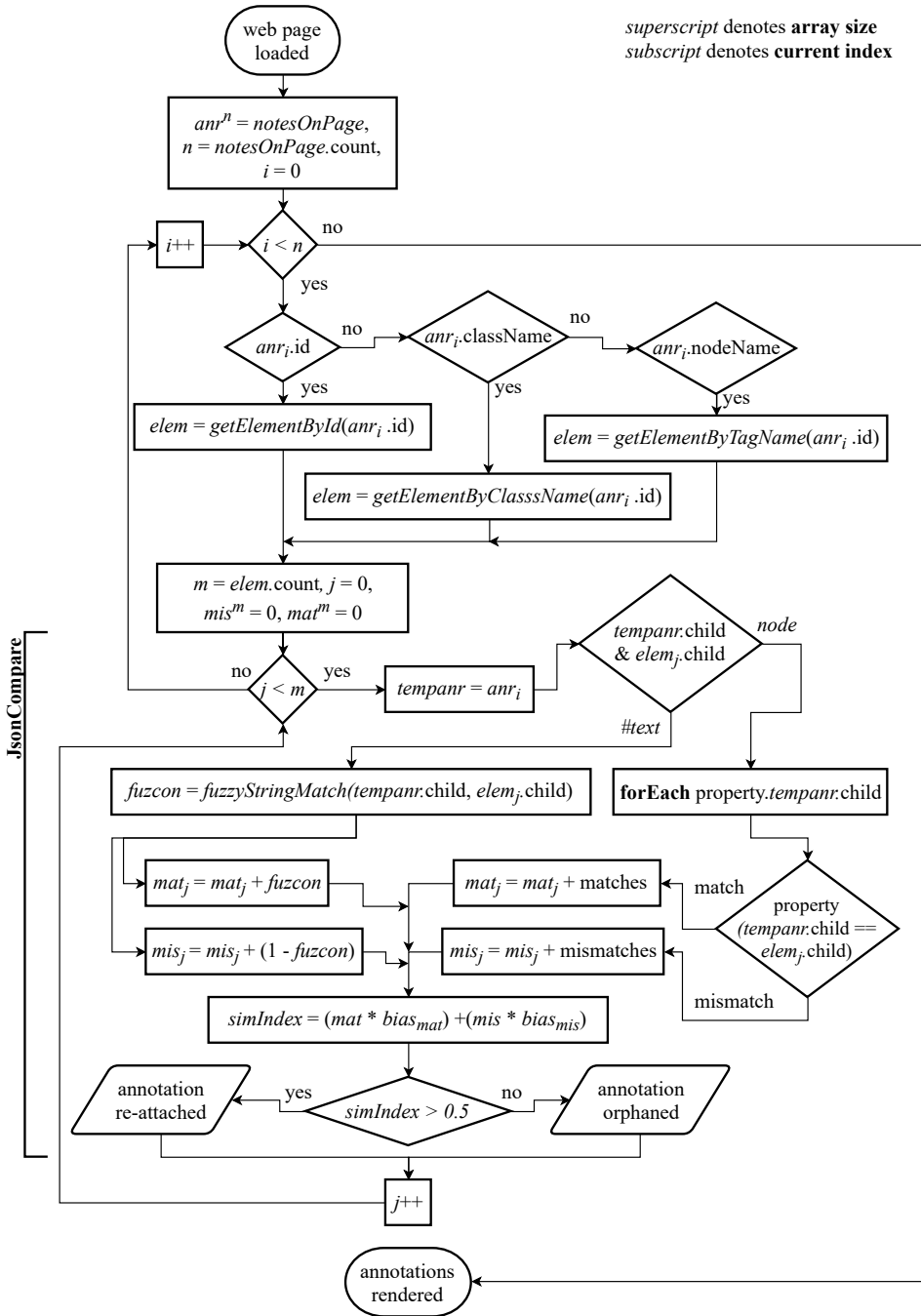


Figure 11 – Flowchart illustrating the proposed DOM-Oriented Edit-Distance anchoring algorithm. [publication II].

4.2.2 Experimental Setup and Results

Following the approach used by Brush et al. [45], to evaluate the proposed algorithm, we first designed a novel web annotation system (see Chapter 5) around the proposed

algorithm. The fuzzy anchoring approach proposed by Hypothes.is is the latest among the current state-of-the-art; therefore, we decided to compare the proposed algorithm against Hypothes.is. To achieve this, we examined more than 1000 open (publicly viewable) annotations created on Hypothes.is. To ensure that our intrinsic biases (for instance, when creating annotations) did not have an effect on the final results of the experiment, we selected annotations that were created by regular Hypothes.is users (i.e., the users were not known to us). We then attempted to replicate the identified annotations using our novel web annotation system, Tippanee. We visited the annotated web pages and created the exact same annotations (as the ones we examined on Hypothes.is) using our annotation tool. In total, we were able to replicate 735 publicly viewable (at least at the beginning of the experiment) annotations. We then left these annotations aside for a period of more than 30 days. The premise behind this action was that some of the annotated web pages would decay overtime, and if that happens, we should be able to compare the robustness of the proposed algorithm against Hypothes.is' fuzzy anchoring algorithm. On revisiting the annotations after the designated period, we found that that Tippanee's anchoring algorithm, was able to generate similarity indexes for 675 annotations. And among these 617 annotations (i.e., 91.41%) were successfully reattached. For the current experiment, the threshold for orphaning, i.e., the similarity index below which an annotation would be considered orphaned was set at 50%. Among the reattached annotations, 538 annotations (i.e., 88.0%) returned similarity indexes of more than 0.9, i.e., the texts the annotations were reattached to were more than 90% similar to their corresponding annotated texts.

Unfortunately, when searching for the selected annotations on Hypothes.is, it was found that a large number of annotations were already deleted by their original authors (or, were possibly changed from public to private view), thereby preventing us from verifying the quantity of annotations that were re-attachable on Hypothes.is. In hindsight, it would have been better to create the selected annotations on Hypothes.is using a personal account on the platform as well, as that would have allowed us to come back to the annotations at the end of the experiment; and thus, allowed us to compare the performance of Tippanee's anchoring algorithm against Hypothes.is, on an equal setting. That said, reverting to Aturban et al.'s work [26], we assumed that in the worst-case scenario at least 72.7% of the annotations created on Hypothes.is should have been re-attachable, and based on this proposition, we derived an initial estimate that the proposed anchoring algorithm was able to successfully reattach at best 18.71% (i.e., 91.41% *minus* 72.7%) more annotations than Hypothes.is.

Figure 12 presents the number of reattached (indicated in *blue*) and orphaned (indicated in *red*) annotations, as examined on Tippanee. Out of the 675 annotations studied in the above-mentioned experiment, the proposed DOM-Oriented Edit-Distance anchoring algorithm (integrated into Tippanee) was able to successfully reattach 617 annotations; that is to say, that 91.41% of the studied annotations were reattached with similarity indexes more than 50% (considered to the second decimal), while the remaining 58 annotations returned a similarity index of less than 50% and were therefore marked as orphaned. The numerical values indicated at the top of each of the bars (in Figure 12) represent the number of annotations in their corresponding similarity index range; for instance, the similarity index range of 538 annotations (on the extreme right) was found to be more than 90% and less than equal to 100%, similarly, the similarity index range of 27 annotations (on the extreme left) was found to be more than 0% and less than equal to 10%.

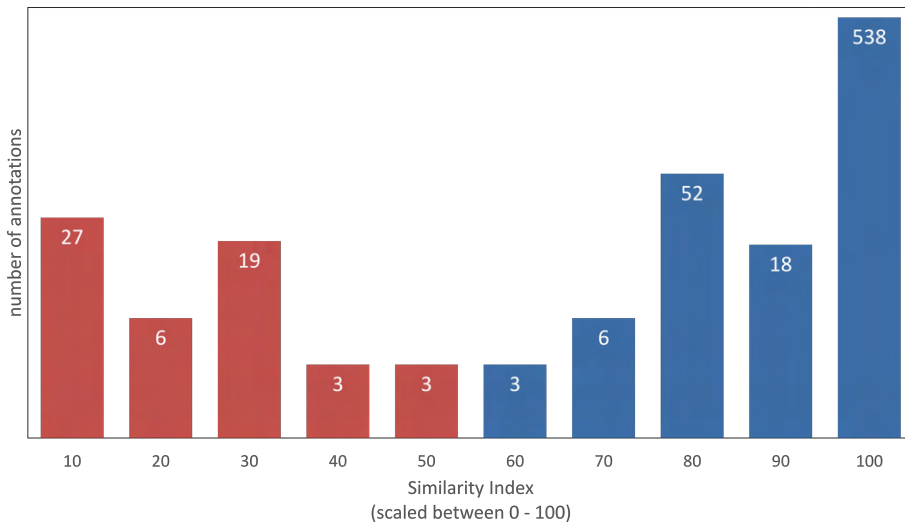


Figure 12 – Similarity indexes of orphaned (in red) and reattached (in blue) annotations studied during the evaluation of the proposed DOM-Oriented Edit-Distance anchoring algorithm. adapted from [publication II].

4.2.3 Discussion

Given the complex and unpredictable nature of web page decay, we acknowledge, that comparing anchoring algorithms over a small data-set of a few hundred or thousand annotations does not empirically and irrefutably validate the robustness of any anchoring algorithm. Furthermore, considering that anchoring algorithms and web annotations have only recently re-emerged (i.e., regained interest) in research communities, it could be argued that there is a lack of the fundamental knowledge and understanding in the current pool of scholarly literature, when it comes to the evaluation of robustness of anchoring algorithms. As we alluded to in Chapter 2, though several annotation systems have been developed in the past, most such systems have only been evaluated with focus on usability and user-friendliness. Hence, guidelines and recommendation suggesting approaches for methodological evaluation of such systems and their underlying technologies, has not emerged in due time. Looking at the current body of knowledge, we find that the evaluation methodology used in this work is sufficient to prove that the proposed algorithm is (to some extent) better than the state-of-the-art, but possibly only in some respects (i.e., for particular web page texts and structures).

It is important to note here that, during the presented evaluation we did not utilize any web archival solutions, neither attempted to reattach orphaned annotations using archived (annotated) web pages. Although as demonstrated by Aturban et al. [26], Hypothes.is allows its users to reattach annotations to archived versions of annotated web pages, this however, is not possible when using Tippanee. This is because, while Hypothes.is reattaches annotations on its server-side, Tippanee does so on the client-side (see Section 5 for details); and since the URI of an archived web page is different from its originally annotated URI, Tippanee is not able to recognize the embedded URI, thereby preventing it from recognizing that the annotation belongs to said archived web page. Considering these limitations, we acknowledge that to irrefutably validate our findings, we need to conduct further examination and experimentation. To this end, in Chapter 6, we present a first-of-its-kind, test-bench for evaluation of anchoring algorithms.

4.3 Summary and Conclusions

To summarize, in this chapter, we examined the state-of-the-art in web anchoring algorithms. We first delved into the various anchoring algorithms used by state-of-the-art web annotation systems and highlighted their drawbacks. Design choices and approaches that could be integrated into a novel anchoring algorithm, were also identified. We also presented arguments supporting our observation that, the likelihood of an annotated web page changing textually is much higher than its chances of changing structurally. Based on our findings, we concluded that a more robust and accurate anchoring algorithm must take into account the structure of annotated content. And thus in regards to the research question RQ2.1, it can be concluded that textual web annotations can be prevented from being orphaned, by using an anchoring algorithm that not only focuses on annotated texts, and its surrounding context, but also, on the annotated contents' HTML DOM structure. Given the lack of such an anchoring algorithm in literature, we proposed an novel algorithm that utilizes an annotated content's text and it's structure, to robustly attach annotations to their correct locations, even when the annotated web pages have decayed. The algorithm combines components from state-of-the-art anchoring algorithms, with a novel DOM-oriented annotation matching approach. The proposed algorithm was evaluated by replicating 1000 Hypothes.is annotations (created by real web-users), into a novel web annotation system (described in Chapter 5), designed around the algorithm. At the end of the experiment, we found that the proposed algorithm performed more than 18.71% better than the current state-of-the-art algorithm.

The primary limitation of the evaluation presented in this chapter, is the small number of textual annotations that the proposed algorithm was tested on. We address this issue, together with RQ2.3, in Chapter 6, where we introduce a novel test bench we designed to empirically evaluate the robustness and accuracy of web annotation system's anchoring algorithms.

5 The Web Annotation Platform Tippanee

The work presented in this dissertation has been developed on the empirically validated proposition that web annotation systems are effective tools when it comes to knowledge creation and sharing, as has been illustrated in literature [160, 97, 152]. However, these systems are conventionally marketed as tools that enable web users to highlight, comment, store and share annotations over web documents. And as suggested in literature, this has led to concerns such as whether features provided by web annotation systems are unique enough to motivate users to use them during learning. As argued by Kalboussi et al. [97], many activities related to learning and knowledge sharing are already enabled by discussion forums, wikis and other similar platforms. We agree with this assertion, and find that web annotation systems can be improved as a whole by integrating Anderson's ideas on preserving web content [22], also by bringing in new feature sets that are conventionally not possible in state-of-the-art web annotation systems in part due to the systems' design and underlying algorithms.

To this end, this chapter presents a novel web annotation system, built around the (previously proposed) novel anchoring algorithm, presented in Chapter 4. In this chapter, we first briefly discuss annotation activities supported by commonly used web annotation tools, and then identify additional annotation activities that could support knowledge co-creation. With the aim to fill the identified feature gaps, we present a novel annotation platform that addresses these pitfalls. The work presented in this chapter addresses the research question RQ2.2:

“How to design a stable web annotation platform that does not fully rely on content providers?”

The work presented in this chapter is part of Thesis Contribution C3, proposed in **publication II**.

5.1 Conventional Web Annotation Activities

As discussed in Chapter 2, among the several web annotation systems that have been developed over the years, only a handful are currently in use today. These are Diigo [11], Annotate [2], Genius [4], and Hypothes.is [7]. Among these, Hypothes.is has the most active users and community support. Also, as stated in Chapter 2.2, services provided by Diigo, Annotate, and Genius can only be accessed by users once they log in on to the service; whereas, annotations created on Hypothes.is can be viewed without signing up or logging in to the service. Finally, Hypothes.is' code is open source, while Diigo, Annotate, and Genius' is not. However, irrespective of the annotation systems' popularity, all of the above mentioned tools offer similar sets of system features, although with slight differences. The features these systems offer include: the ability to annotate text by highlighting, adding comments, storing annotations (for later usage), and the ability to share annotations with other system users. Please note that, when referring to system features or features, we adhere to the ISO/IEC/IEEE International Standard (systems and software engineering) [20] definition of 'software feature' which is defined as “software characteristic specified or implied by requirements documentation [...] example: functionality, performance, attributes, or design constraints” [20].

Highlights The action of highlighting text when using web annotation systems is similar to the physical action of highlighting text in a book (or other similar written or printed works) using a highlighter or pen. The primary aim of the action is to create a distinction between the highlighted text and the remaining work, in such a way that

when the whole text is viewed, the readers are drawn to the text that has been highlighted.

In the same way, the highlighting feature of web annotation systems, allows users to demarcate parts of textual web content. Once highlighted, users can revisit the highlighted web pages again at a later time, and then can (ideally) still find the highlighted text on the web page. This is achieved by storing the anchor associated to the highlighted text within the web annotation system (typically on the server-side). Apart from viewing the highlighted text, on the host web page, users can also view the stored highlighted texts within the web annotation system's dashboard.

Comments This action is made up of two parts, highlighting and commenting. The activity is an extension of highlighting and requires users to add some textual comment to the highlighted text. The comments made here are meant to provide some explanation to the highlighted text, as is the case with physical annotations. Comments are paired (i.e., permanently linked) to the anchors generated from the highlighted text.

Chats via. comments Comments added by users can also act as a means for communication (i.e., chat). By adding comments to other comments (associated to the same highlights), to-and-fro, two or more users can engage in conversation while using web annotations. This action is similar to how web users can chat using comments on blogs and threaded forums (and other similar web applications such as Quora or Stack Overflow).

By allowing users to add comments to highlighted texts, web annotation systems allow users to carry out more focused discussions [173]. These discussions provide users with experience that is distinct from discussions on threaded forums, or discussions carried out while using chat apps and emails. This is because, on discussion forums, discussions often tend to go off-topic, while when using chat apps and emails, the topic of discussion (for example, some text) and the discussion itself happens on different platforms. For example, the topic of discussion could be an article on Wikipedia, while the discussion itself would typically happen in a chat room or over emails. In both cases, the topic (being discussed) is on a different platform while the discussion happens on another. When chatting using web annotations, however, discussions can be carried out directly over web pages with the content tied to the topic of discussion.

Storing annotations Annotations created on (state-of-the-art) web annotation systems are stored in data silo (i.e., on the server-side). These annotation systems are web applications; therefore, users can only access their annotations once they are connected to the annotation system's server, either through the annotation system's website or its browser extension. In both cases, once the users log in on to the service, they can access their annotations and highlights, either on the web annotation system's dashboard, or by visiting the annotated web page through the annotation system.

Searching annotations Users can search for specific annotations (created by them, or shared with them), using either the keywords of the annotated texts (and their respective comments), or through tags (in cases like Hypothes.is). These tags, are typically single-word classifiers added by users themselves, to make it easier to browse through multiple annotations.

Sharing annotations Finally, the action of sharing annotations can be carried out in multiple ways. Users can either create (user) groups and invite other users to collaboratively create annotations; or, they can share annotations (only in Hypothes.is) by sharing the URI indicating the annotated text. The system feature that enables sharing annotations is key to communication, as users can only comment over other users' annotations, if the annotations have been shared with them.

As the system features described above are standard across all currently used web annotation systems, a novel annotation tool must include these features. However, as stated in Chapter 3.3, if web annotation systems are to evolve to be more useful in knowledge co-creation, they must include additional features to support knowledge exchange. Unfortunately, such features are still currently being investigated, and therefore, the specifics of these features are not known. Furthermore, as pointed out by Sun et al. [173], current research on the use of web annotations in learning and education has primarily focused on the impact of annotations on learning outcomes, and not too much on learning processes. Therefore, identifying system features that might support learning and knowledge exchange processes requires further exploration and experimentation.

5.2 Artifact Development

Building on these (web annotation) system features, and with the aim of evaluating the proposed anchoring algorithm (described in Chapter 4), we designed a novel web annotation tool that we call Tippianee. The design decisions considered during the development of the web annotation tool were informed by the motivations drawn out in Chapter 2. Furthermore, additional attention was paid to user-centric requirements such as data ownership and ease of use.

The proposed artifact is designed to work primarily as a stand-alone tool, and therefore does not rely on a server for storing annotations and reattaching anchors (unlike the state-of-the-art systems). However, in order to support collaboration, the system does include server-side functionalities. The tool is designed as a lightweight Google Chrome browser extension and is freely available on the Google Chrome Web Store [142, 141]. As Tippianee is designed to function without relying on server-side computation, the (proposed) anchoring algorithm is built directly into the tool's browser extension. This means that Tippianee's users can create, reattach and access annotations without signing up to or logging in on to the server. This further implies that, when users use the tool in stand-alone mode (i.e., without connecting to the server), the annotations created by the system's users are stored directly on the users' local machines (and not the server-side). This empowers the system's users, as the users have complete ownership of their data, and thus there is no risk of data theft or data leakage. The client side of the web annotation tool (i.e., the browser extension) is designed using HTML, CSS, Vanilla Javascript (JS), and jQuery.

In order to make sure that the annotation tool is easy to use, the user interface (UI) of the tool is heavily influenced by the design of Hypothes.is' UI. Similar to Hypothes.is, Tippianee's UI shows up on the right side of the screen (see Figure 14). Following Hypothes.is' design, Tippianee too presents itself as layer over web pages. This is achieved by adding the HTML, CSS, and JS code for the annotation tool's UI into the web page (being viewed by the users), after the page has loaded onto the browser. Here, a key difference is that, in case of Hypothes.is, the annotation tool's UI is generated on the server-side (and then added through APIs), whereas, in case of the Tippianee, the process of adding the annotation tool's UI is carried out on the user's local machine (on which the web page is being

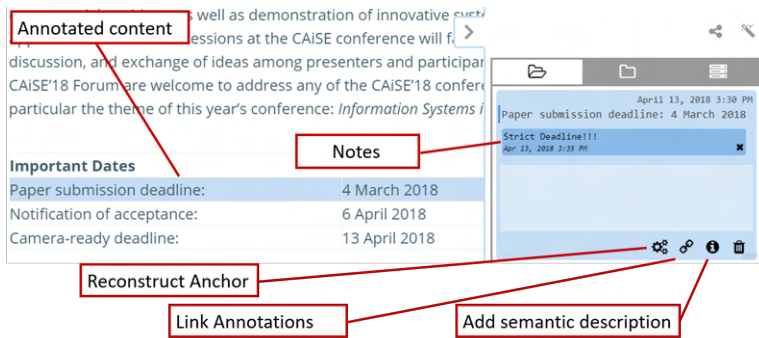


Figure 13 – Illustration of Tiptanee User Interface. adapted from [publication II].

viewed). Once the annotation tool’s UI is successfully added, users can then utilize the systems features (i.e., create highlighted annotations and more).

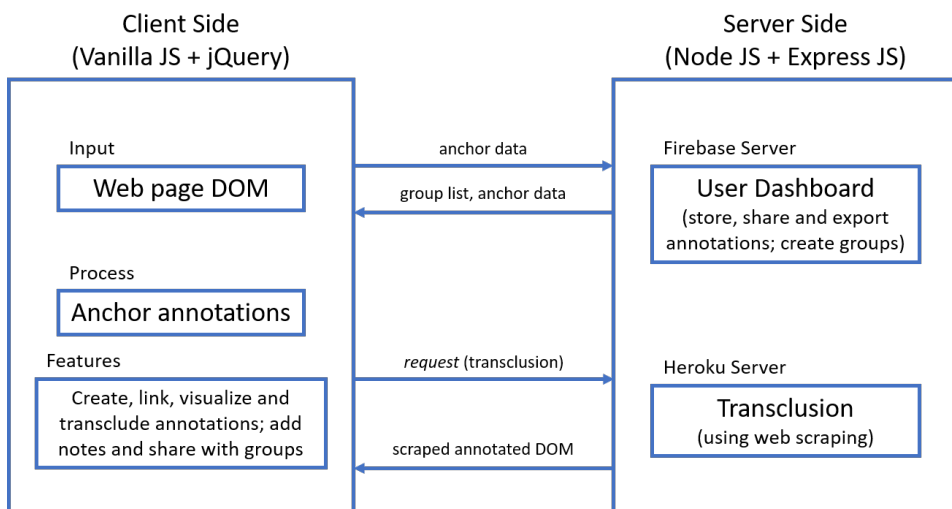


Figure 14 – Illustration of Tiptanee’s System Architecture. adapted from [publication II].

Tiptanee’s server-side functionalities are designed using Node JS and Express JS. If users opt to share or export annotations, or create user groups, they are required to sign in to the service. Using a server for sharing annotations is necessary, as annotations and comments shared within groups have to be synchronized constantly among group members. Alternatively, there are other solutions for exchanging data over web browser (for example, peer-to-peer data exchange like the the one that we explored in **publication III** and **publication IV**), however, in order to keep the development process simple, we decided to utilize the conventional method of using servers. By working together with the system’s server-side features, Tiptanee’s browser extension enables its users to carry out additional annotation-related activities (as illustrated in Figure 14). The subsystems of Tiptanee corresponding to the anchoring algorithm are illustrated in Fig. 15.

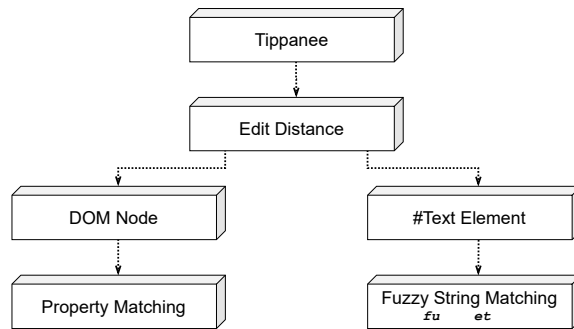


Figure 15 – Architecture of Tippianee’s anchoring subsystems. adapted from [publication II].

5.3 Novel System Features

The anchoring algorithm underlying Tippianee relies on both textual content and HTML DOM structures to generate anchors; this allows for creation of new system features that can not be achieved when using simple text-matching anchoring algorithms (for example, like keyword-searching or string-matching).

5.3.1 Local Storage

As Tippianee’s browser extension is primarily designed to be used as stand-alone system, the tool’s storage functionality utilizes the browser’s (in the current case, Google Chrome’s) internal storage system. Using the ‘chrome.storage’ API, Tippianee is able to store and retrieve annotation data from within the browser. Although using the browser’s internal storage has its own drawbacks, for example, resetting or uninstalling the browser can lead to loss of annotations; it also provides multiple advantages. For instance, since the web annotation tool is designed for Google Chrome browsers, users can synchronize their annotations over multiple browsers across different operating systems and machines. Furthermore, given that Tippianee also has a server side, users can use the tool’s server storage as a fallback mechanism. Additionally, users can also choose to store annotations both offline (i.e., locally) and online (i.e., on the server). This is the only one of the features, that makes Tippianee, distinct from other state-of-the-art web annotation systems. Also, we find that this system feature is critical as it addresses the concerns of data ownership and reduces the user’s dependence on web annotation service providers.

5.3.2 Reconstructing Anchors

Since Tippianee’s anchoring algorithm stores both textual content and DOM structures when generating and storing anchors, the system is able to preserve annotated content in their original states. The various DOM attributes that the system’s anchoring algorithm uses to detect and track anchors can also be utilized to reconstruct anchors (together with both the annotated content and its surrounding context) in their original form. For example, consider the scenario that a user created an annotation. Over time, due to web page decay, the annotated content changed. Now, in conventional web annotation systems, the annotated content could be orphaned. If this happens, the user would only be able to view the annotated text, as is in Hypothes.is, even though the annotation system’s anchoring algorithm stores 32 characters before and after the annotated text. In case of Tippianee, the generated anchor has enough information about the annotated text, so that it can convert the anchor back into its HTML DOM form, this is enough to demonstrate

to the user, how the annotation looked like (with respect to textual content) when it was created. Again, this system feature only becomes possible by the tool's novel anchoring algorithm. This feature is particularly unique, as it not only enables one of Anderson's recommendations for data preservation [22], but also ensures that the user's annotations are stable even after web page decay. The feature is also critical with respect to knowledge exchange as by empowering users with stable annotations, the system ensures that the knowledge shared through the annotated text is not lost over time.

5.3.3 Similarity Index

Enabled by the novel anchoring algorithm, Tippane provides its users with a visual aid indicating the amount of changes to the annotated content. As explained in Chapter 4.2.1, the proposed algorithm generates a numerical value (referred to as the similarity index) that indicates how much the annotated text has changed since it was created. This index can assist users in estimating the *ephemerality* of the annotated web page and thereby prompting them to annotate web pages that are less likely to change frequently. The exact benefits of this feature have not yet been explored, however based on literature, a potential application of this feature could be to track web pages changes, similar to change detection and notification tools [118].



Figure 16 – Illustration of Tippane's Similarity Index feature.

5.3.4 Linking Annotations

Another novel feature offered by Tippane is the ability to link web annotations to each other. Although the W3C's WAD model has frameworks supporting annotation linking, the feature is currently lacking among all conventional web annotation systems. Made possible by the tool's anchoring algorithm, linking annotations enables users to better organize and structure their created annotations. Furthermore, by allowing users to visualize linked annotations as graphs similar to information/knowledge graphs, the system is able to support knowledge creation and exchange in a manner that is better than exchanging conventional (independent) textual annotations. Figure 17 illustrates linked annotations, as an annotation graph within the Tippane's UI.

5.3.5 Transclusions

Finally, Tippane's anchoring algorithm enables its users to make use of transclusions. First proposed by Nelson in the 1980, in his work 'Literary Machines' [128, 129], transclusion is the process by which content from one source (or document) can be displayed (and referenced) onto a second source. Transclusions are commonly used in Excel spread-

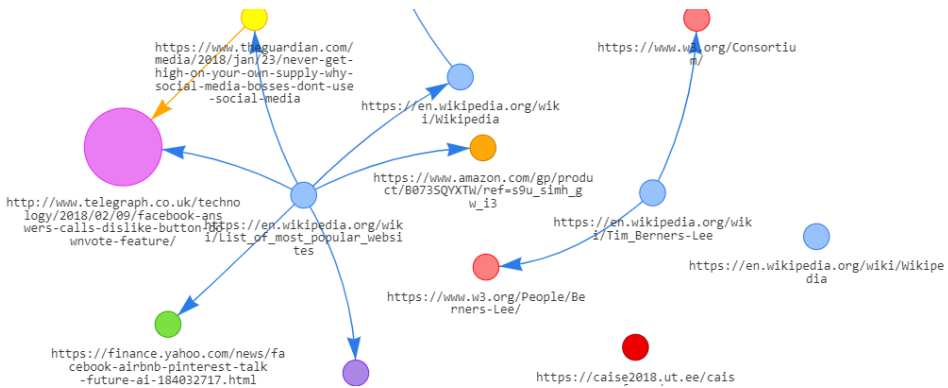


Figure 17 – Annotation graphs as viewed on Tippanee.

sheets, where data stored in one table cell can be made visible in another table cell, even across several excel sheets.

In the context of web annotations, Tippanee allows its users to view the current state of an annotation created on one web page, on to a different web page. In order to utilize this feature, Tippanee users are required to log in to the service. When a user transcludes an annotation from one web page to another, the server first accesses the host web page. The tool then finds the annotation using its anchoring algorithm, and inserts the DOM element of the annotated element into the page where the transclusion has been requested. Using this feature allows Tippanee users to transclude (i.e., access and view) annotations from one web page to another. We argue that this feature further supports knowledge exchange, as it prevents users from hopping around multiple data sources (or browser tabs) and therefore ensures that all discussions and their respective (topic/data) sources can be accessed on the same browser tab.

5.4 Lab Experiment

To evaluate the usability and usefulness of the developed web annotation system, we conducted a lab experiment, through which we sought to understand how easy it was for a first-time user to use the web annotation system Tippanee. Through the experiment we also aimed to identify the potential applications of the proposed tool, especially given its novel feature sets.

To understand the usability of the system, we recruited 25 participants using both (traditional) face-to-face and online communication including emails and Facebook. The participant group comprised of both men and women between ages of 25 and 45 years. The participants included 12 web developers and 13 master students. The experiment started with a short tutorial on how to use the web annotation system. Following this, the participants were asked to use the tool to create and share as many annotations as possible, over the duration of three hours. During the initial experiment, most participants were able to create between 50 and 100 textual annotations. After three hours, the participants were asked to continue using the annotation tool over the next seven days. The participants were also asked to try and use the tool for knowledge and information exchange in their natural working environments. After 7 days, participants who were still interested in actively using the web annotation tool were given seven more days to further explore the tool.

Fourteen days after the initial experiment, the participants were asked to fill a short

questionnaire including questions that had to be answered through Likert scales, some open-ended questions regarding their experience with the tool, and additional space for providing feedback on the tool's shortcomings and systems bugs. The question included in the questionnaire were as follows:

- Q1** How likely is it that you would recommend Tippanee to a colleague? [On a scale of 1 to 10]
- Q2** How satisfied are you with Tippanee's ease of use?[On a scale of 1 to 5]
- Q3** How satisfied are you with the look and feel of Tippanee's Google Chrome extension? [On a scale of 1 to 5]
- Q4** How satisfied are you with the account setup experience of Tippanee's dashboard? [On a scale of 1 to 5]
- Q5** How satisfied are you with the reliability of Tippanee's anchoring approach? [On a scale of 1 to 5]
- Q6** How satisfied are you with the ability to collaborate with other users on Tippanee? [On a scale of 1 to 5]
- Q7** Which kind type of websites did you annotate during the workshop? [Choose one or more from: Blogging, Community building, Education, News, Search engine, Social networking, and E-Commerce]
- Q8** What would you like to use Tippanee for? [Choose one or more from: Expression of opinion, Information sharing, and Social interaction]
- Q9** How likely is it that you will use Tippanee after the experiment? [On a scale of 1 to 10]

5.5 Results and Discussion

The participants' response to the questionnaire are presented in Tables 3, 4, 5, and Figures 18 and 19. Table 6 and Figure 20 demonstrate the distribution of the participants' responses to the questions Q7 and Q8, and maps the relation between the individual choices provided under both questions.

Table 3 – Summary of participants' response to the questions Q1 - Q6, and Q9.

	Q1	Q2	Q3	Q4	Q5	Q6	Q9
Valid	25	25	25	25	25	25	25
Missing	0	0	0	0	0	0	0
Mode	10.00	4.00	4.00	4.00	4.00	4.00	10.00
Median	9.00	4.00	4.00	4.00	4.00	4.00	8.00
Mean	8.24	3.80	3.96	4.12	3.92	3.96	7.96
Std. Deviation	2.17	0.87	0.89	0.83	0.95	1.06	2.37
Range	8.00	3.00	3.00	3.00	3.00	4.00	9.00
Minimum	2.00	2.00	2.00	2.00	2.00	1.00	1.00
Maximum	10.00	5.00	5.00	5.00	5.00	5.00	10.00

Finally, the Table 7 and Figure 21 present the correlation between participants' response to the question Q1 - Q6 and Q9.

Table 4 – Summary of participants' response to the question Q7

Q7	Frequency	Percent	Valid Percent	Cumulative Percent
Blogging	16	9.76	9.76	9.76
Community building	18	10.98	10.98	20.73
Education	33	20.12	20.12	40.85
News	26	15.85	15.85	56.71
Search engine	27	16.46	16.46	73.17
Social networking	25	15.24	15.24	88.41
E-Commerce	19	11.59	11.59	100.00
Missing	0	0.00		
Total	164	100.00		

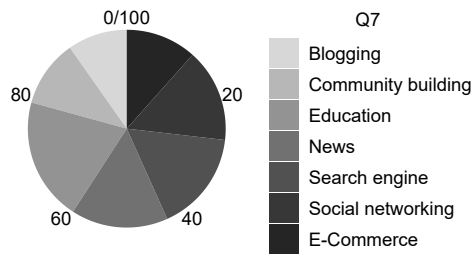


Figure 18 – Pie chart illustrating the distribution of participants' response to the question Q7.

Table 5 – Summary of participants' response to the question Q8

Q8	Frequency	Percent	Valid Percent	Cumulative Percent
Expression of opinion	36	21.95	21.95	21.95
Information sharing	92	56.10	56.10	78.05
Social interaction	36	21.95	21.95	100.00
Missing	0	0.00		
Total	164	100.00		

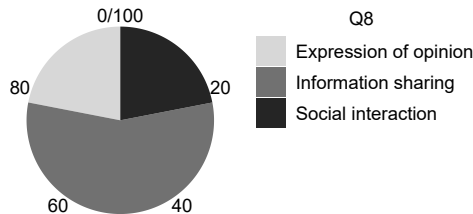


Figure 19 – Pie chart illustrating the distribution of participants' response to the question Q8.

Table 6 – Contingency table mapping the relationship between participants' responses from questions Q7 and Q8.

Q7 ↓ / Q8 →		Expression of opinion	Information sharing	Social interaction	Total
Blogging	Count	3.00	10.00	3.00	16.00
	% r	18.75%	62.50%	18.75%	100.00%
	% c	8.33%	10.87%	8.33%	9.76%
Community building	Count	4.00	9.00	5.00	18.00
	% r	22.22%	50.00%	27.78%	100.00%
	% c	11.11%	9.78%	13.89%	10.98%
Education	Count	7.00	19.00	7.00	33.00
	% r	21.21%	57.58%	21.21%	100.00%
	% c	19.44%	20.65%	19.44%	20.12%
News	Count	5.00	16.00	5.00	26.00
	% r	19.23%	61.54%	19.23%	100.00%
	% c	13.89%	17.39%	13.89%	15.85%
Search engine	Count	7.00	13.00	7.00	27.00
	% r	25.93%	48.15%	25.93%	100.00%
	% c	19.44%	14.13%	19.44%	16.46%
Social networking	Count	6.00	13.00	6.00	25.00
	% r	24.00%	52.00%	24.00%	100.00%
	% c	16.67%	14.13%	16.67%	15.24%
E-Commerce	Count	4.00	12.00	3.00	19.00
	% r	21.05%	63.16%	15.79%	100.00%
	% c	11.11%	13.04%	8.33%	11.59%
Total	Count	36.00	92.00	36.00	164.00
	% r	21.95%	56.10%	21.95%	100.00%
	% c	100.00%	100.00%	100.00%	100.00%

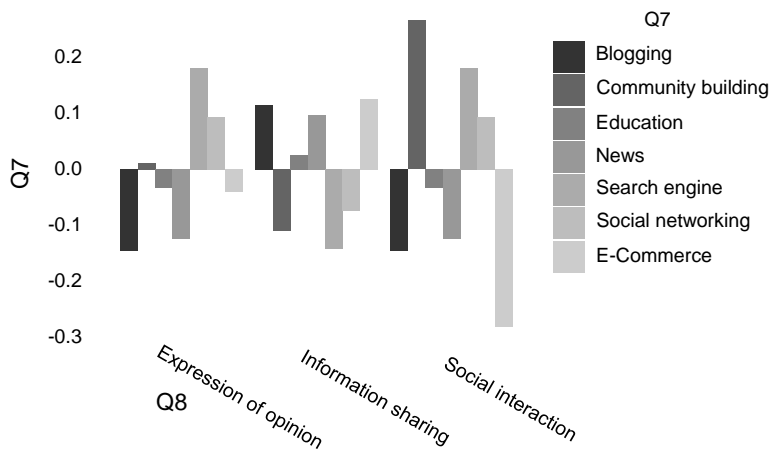


Figure 20 – Graph illustrating the relationship between participants' responses from questions Q7 and Q8.

Table 7 – Pearson's correlation between participants' response to the question Q1 - Q6 and Q9.

		Pearson's r	p
Q1	- Q2	0.78***	3.89e-6
	- Q3	0.83***	3.30e-7
	- Q4	0.58**	2.17e-3
	- Q5	0.66***	3.80e-4
	- Q6	0.78***	3.38e-6
	- Q9	0.92***	9.26e-11
Q2	- Q3	0.80***	1.50e-6
	- Q4	0.79***	3.24e-6
	- Q5	0.74***	2.71e-5
	- Q6	0.76***	9.34e-6
	- Q9	0.81***	1.07e-6
Q3	- Q4	0.68***	1.72e-4
	- Q5	0.73***	3.06e-5
	- Q6	0.66***	3.16e-4
	- Q9	0.83***	2.96e-7
Q4	- Q5	0.69***	1.17e-4
	- Q6	0.76***	9.99e-6
	- Q9	0.64***	6.40e-4
Q5	- Q6	0.66***	3.69e-4
	- Q9	0.66***	3.16e-4
Q6	- Q9	0.75***	1.91e-5

* $p < .05$, ** $p < .01$, *** $p < .001$

Overall most participants reported that they found the annotation tool was both easy to use, and useful. A majority of the participants felt that the annotation tool assisted them by enabling better information sharing (as compared to conventional methodologies). Many participants also stated that the tool was especially useful for social interactions, sharing interesting topics and for expressing opinions. Participants suggested that they mostly used to tool to annotate content on educational and news websites, Q&A portals, and search engines.

Apart from the conclusions that were drawn directly from the participant's responses, we also found additional interesting correlations between the participant's intentions and choices while using the Tippane web annotation tool. As illustrated in Figure 20, we found that participants who used to tool for 'expression of opinions' preferred to annotate search engine pages and social networking websites. Whereas, participants who used Tippane for 'information sharing' were more likely to annotate blogs, news websites and e-commerce platforms. Finally, participants who choose to use the tool for 'social interactions' preferred to annotate community building websites, search engines and social networking platforms. Additionally through Table 7 and Figure 21, we found that participants who planned on using Tippane after the experiment, were also more likely, to recommend the tool to their colleagues. This to some extent, is also clearly visible through the number of users the platform has gained since it was first officially launched in 2018 (as illustrated in Figure 22).

It is important to note here that, since Tippane anchoring algorithm stores both annotated content and its context (i.e., surrounding text and structure) on the client's machine,

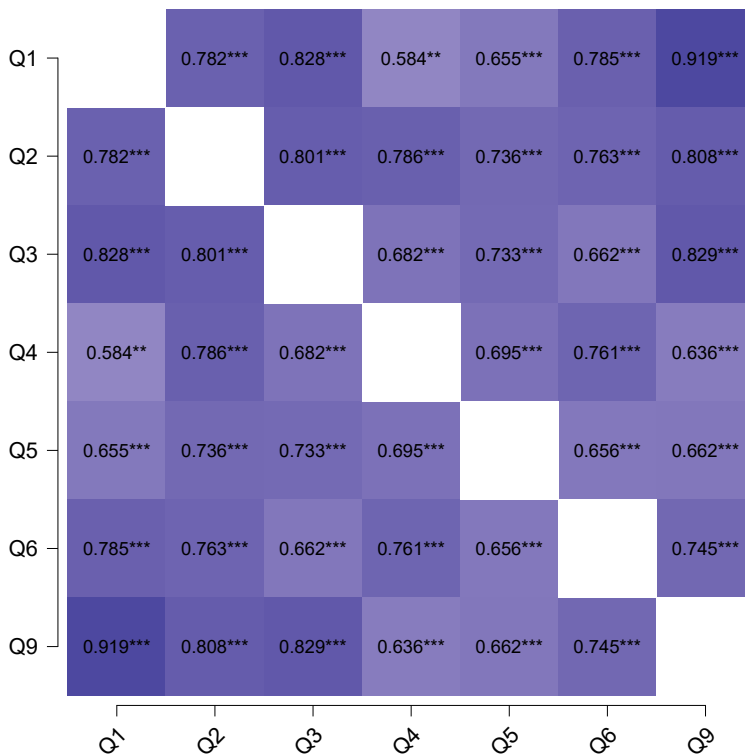


Figure 21 – Heat-map illustrating Pearson's correlation between participants' response to the question Q1 - Q6 and Q9.

there are potential legal implications (specifically with respect to web scraping and content re-use) that need to be considered. However, as we have explained in **publication II**, these concerns have been discussed with an expert on copyright infringement and piracy acts. Furthermore, we also investigated literature on the topic. And based on our investigation, we have concluded that Tipanee, through its features and uniquely thought out system design that supports digital preservation and archiving, does not violate any piracy or copyright laws [**publication II**].

5.6 Summary and Conclusions

The summarize the chapter, in this part of the work, we first explored the system features offered in the state-of-the-art web annotation systems. We then explained the design choices and the thought process behind the development of the novel web annotation tool, Tipanee. Furthermore, we explained the several unique system features introduced by the system and provided insights into how the proposed annotation tool is distinct from other similar systems. We illustrated the tool's architecture and its subsystems, and by doing so concluded that it is possible to design user-centric web tools that do not rely on server-side computation and storage, and that this in turn can extricate web users from fully relying content and service providers (specially in context of web annotation platforms). Thus, answering the research question R2.2.

To evaluate the usability and useful of the proposed web annotation tool, we conducted a lab experiment with 25 participants who used the tool for two weeks. At the

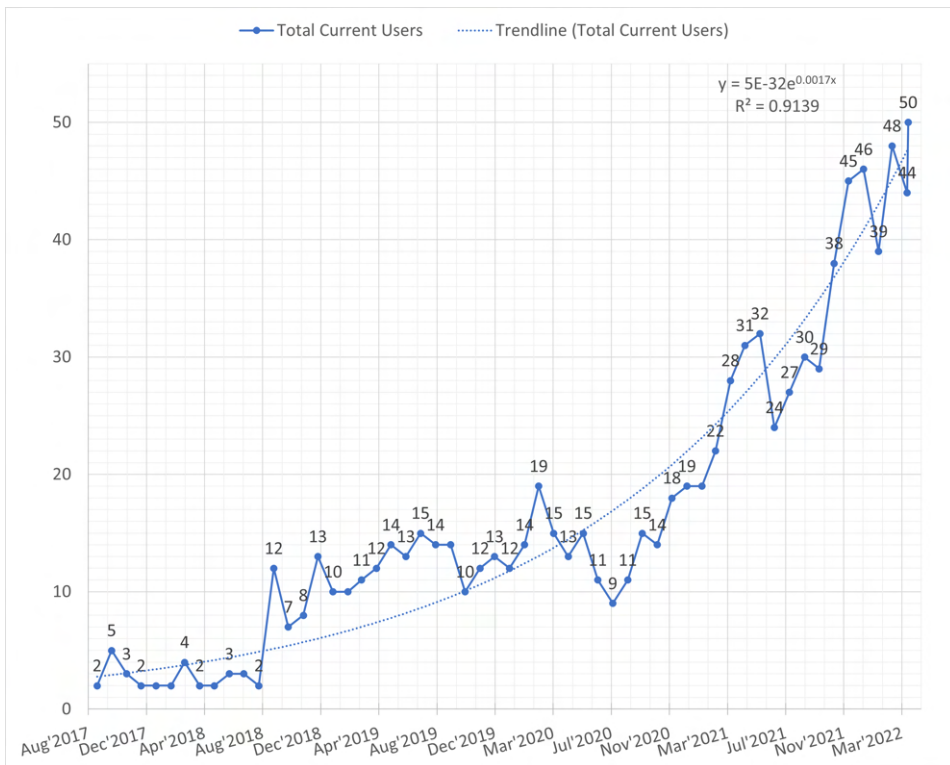


Figure 22 – Graph illustrating growth in Tippanee’s total user count since 2018.

end of experiment we gathered the participants’ feedback through a questionnaire. And by analysing the participants’ responses concluded that most users found Tippanee, both useful and easy to use. We also found some interesting user behaviours, in regards to the users’ motivation behind using annotations and their choice of website types. These findings have been detailed in the chapter through several tables and figures. Finally, in the chapter we briefly mention the legal implications of content re-use in web annotation systems. A detailed explanation of these legal implications have already been presented in **publication II**. Overall, through the proposed annotation tool, we presented a set of novel annotation activities that could potentially support knowledge creation and sharing, when using web annotation tools.

6 Evaluating the Robustness of Anchoring Algorithms

In this chapter, we discuss a web annotation test bench designed to evaluate the robustness of (web annotation) anchoring algorithms. It is the first time, that a test bench for web annotations has been provided. We begin the chapter, by first providing some context about the motivation behind the development of the test bench. We then describe the design and development process of the test bench. Finally, we examine Hypothes.is' state-of-the-art fuzzy anchoring algorithm and compare it against Tippanee's novel DOM-oriented edit-distance approach using the proposed test bench. In particular, through the test bench we address the research question RQ2.3:

“How to evaluate the robustness of anchoring algorithms?”

And by comparing the robustness and accuracy of Tippanee's anchoring algorithm against that of Hypothes.is, we put forward insights that further address the research question RQ2.1:

“How can web annotations be prevented from being orphaned?”

The work presented in this chapter corresponds to the dissertation contributions C2 and C4.

6.1 Overview of the Web Annotation Test Bench

As we pointed out in Chapter 1, among the several challenges being encountered on the web, ephemerality of web content is one that presents the biggest challenge to knowledge creation and exchange. As contents on web pages change over time, they present the risk, that the knowledge stored on these decaying web pages could be lost to future web users. With this in mind, in Chapter 4, we presented a novel anchoring algorithm, aimed at enabling more robust and stable web annotations. However, as we pointed out during the initial evaluation of the algorithm, web page decays, link rots and content drifts are all phenomena, that are well studied in research, however, predicting which pages would decay, and which would not, is still less studied in literature. As suggested by Schneider et al. [167], web content itself is ephemeral and transience in its nature. This implies, that predicting a specific web page's or website's ephemerality is a challenging task. And this means, that evaluating an anchoring algorithm's robustness based on a few hundred or even a few thousand annotations may not be enough to empirically and indisputably validate it. Also, given the ambiguity of the transient nature of web content (and thereby web pages), pinpointing the exact period within which a page would decay is also a daunting task. As suggested in literature, most web pages begin to decay within a few months to a year; however, given that the precise time at which pages start to decay can not be known, creating several thousand annotations and waiting for the annotated text to decay, can also be a challenging task. During our initial experiments with the proposed anchoring algorithm, we revisited the annotations after 30 days, and found that the proposed algorithm performed (at most) 18.71% better than Hypothes.is' anchoring algorithm, however, we acknowledged that the results could vary completely for a different set of annotations or web pages, that is to say, that the results of the experiment are difficult to replicate, and thus, this is an instance of reproducibility crisis.

Given the critical role web annotations play (and will probably more in the future) in e-learning and knowledge exchange, we find that it is critical to develop a benchmark that enable researchers and developers to evaluate the robustness of anchoring algorithms in a reproducible manner. With this in mind, we designed the proposed test bench so that

it simulates textual and structure web page decay based on a predefined set of rules. By creating same annotations on multiple web annotation systems, and by simulating various types and levels of decay on annotated web pages, we argue that one should be able to compare different anchoring algorithms.

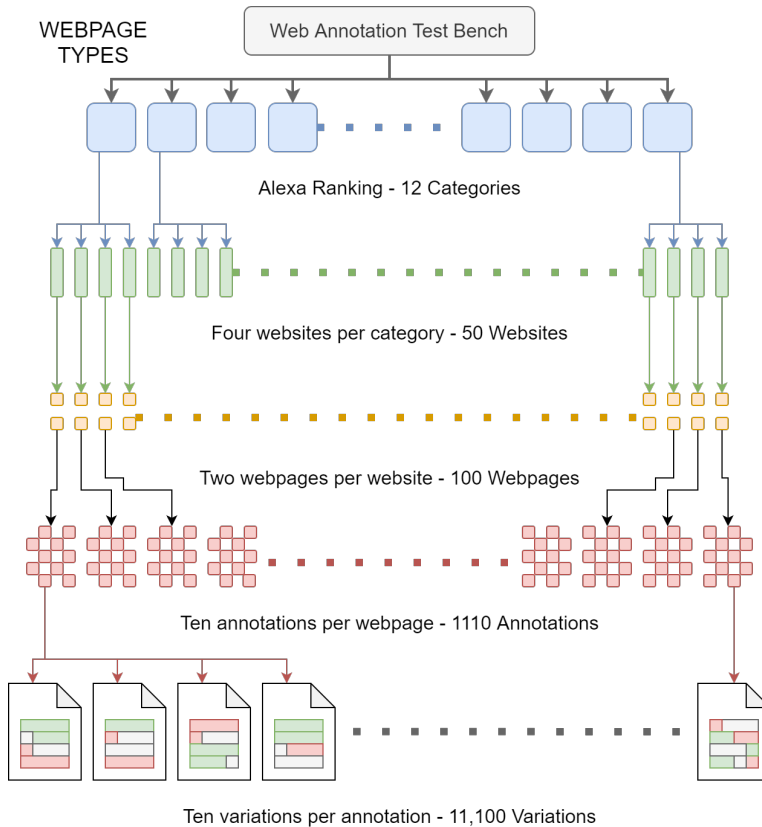


Figure 23 – Illustration of the composition of the proposed test bench.

6.2 Test Bench Setup

The proposed test bench comprises of a collection of web pages acquired from 50 different websites. The websites were chosen from 12 categories, based on Alexa’s top ranking websites in each category. This was done based on the premise that websites with more active users, would have more users interested in creating annotations, and that users may create annotations of all kinds of websites (not only academic or educational). Another key factor for selecting the websites was that typical web users are more interested in creating annotations on websites that are openly accessible (i.e., the ones that do not require users to log in or sign up). Considering these aspects, a good mix of websites, ranging from categories such as arts and news to business, health, sports, and religion, were chosen. Once the websites were identified, the next step was to find and store copies of web pages from these websites. To ensure that each selected web page had at least nine annotations, large pages (with lots of textual content) were selected. Figure 23 illustrates the overall composition of the proposed test bench (including the number of websites, web pages, annotations and their variances).

Each of the selected web pages was then downloaded and stored as a single file. This is different from the conventional method of storing web pages, where each web page has an HTML document, and additional CSS and JS codes in a separate folder. In total, 120 unique web pages were selected and stored for further processing. On each page at least nine annotations were created. In total, 1110 web annotations were created on the selected 120 web pages. It should be noted that of these 1110 web annotations, only 1101 were created on Tippianee, while all 1110 web annotations were successfully created on Hypothesis. The nine annotations that could not be created on Tippianee belonged to a single web page. By simulating 10 different kinds of decays for every single annotation, we were able to achieve 11,100 variances from the 1,110 original annotations (see Fig. 23).

We argue that given the scope of web page decay simulated in the test bench (through the 1,110 original annotations and their 11,100 variances), it should be possible to unequivocally and reproducibly evaluate not only the currently-used anchoring algorithms, but also future ones.

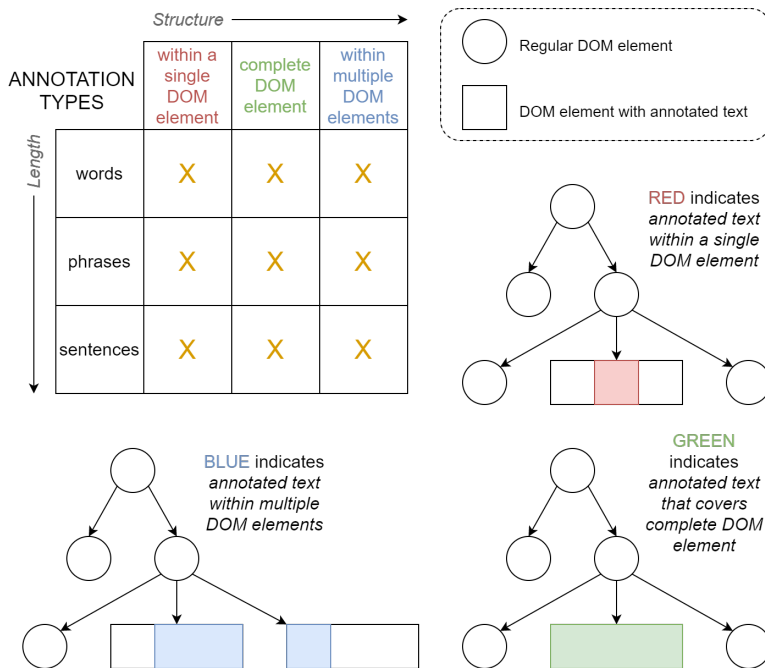


Figure 24 - Illustration of the different annotation types included in the test bench.

6.2.1 Creating Annotations

Based on the controlled experiment from Chapter 4, we found that annotations can be of various types, in particular, based on the length of the annotated text and its position within the web page's DOM structure. When considering the length of annotations, textual annotations can be categorized into three types, namely, words, phrases and sentences; whereas when considering the text's positioning with respect to the DOM, annotations can either belong within a single DOM element, or can cover a complete element. Additionally, annotations can exist at a position where part of the annotated text belongs to one DOM element while the remaining belongs to another. When considering the length of the annotations, we found that smallest annotations are often single words,

and that such annotations can also be numerical values (for example, prices of objects or some statistical number). Phrases on the other hand, can consist of a few words, but not complete sentences. Finally, long sentence-length annotations can be single, or even multiple sentences (for example, verse of a poem or a quote), or can be as long as full paragraphs with 100 or more words (for example, parts of scholarly articles).

With regards to an annotation's position within the DOM, an annotation can either range across a complete DOM element, implying that the all text inside the element is annotated. Or, an annotation can also be positioned 'within' a single DOM element, implying that some text within the said DOM element is annotated, while some text is not. Lastly, an annotation can also extend across multiple DOM elements, which implies that the annotation can contain parts of text from multiple consecutive DOM elements. These annotations can either be long texts or can be multiple smaller-/regular-sized texts (for example, object prices with currency symbols) arranged into a poorly structured DOM.

By combining the different annotation types under both categories, we deduced that web annotations can be of nine types (see Figure 24). Therefore, each web page in the test bench has at least nine different annotations, each belonging to a different annotation category. After finalizing the annotation types, we manually created annotations on the collected web pages, once using the Hypothes.is web annotator, and then using Tippanee. Please note that the annotations created on both platforms were exactly the same, this is important as variance of even a single character can effect the results of the anchoring process (in both tools) and thereby skew the results. In total, the proposed test bench contains 1,110 annotations created on 120 web pages.

6.2.2 Simulating Web Page Decay

Once the 1,110 annotations were successfully created, the next task was to simulate web page decay. As we alluded to in Chapter 6.1, web page decay can occur at both textual level and structural level. We, however, further argue that textual web page decay is more likely to occur on web pages, as the underlying content of dynamic (i.e., DB based) websites can easily be changed by simply modifying the data in their underlying databases. While making structural changes to websites and web pages requires additional time and effort. Also structural changes require lots of planning, as frequent changes to websites structure can be distracting (even, off putting) for the websites' frequent visitors. With this premise in mind, we designed six distinct types of textual changes (see Sect. 6.2.2.1) and four distinct types of structural changes (see Sect. 6.2.2.2) into the test bench.

6.2.2.1 Textual Decay Types As both Hypothes.is' fuzzy anchoring algorithm and Tippanee's DOM-oriented edit distance approach rely on the same fundamental algorithm (i.e., Levenshtein's Edit-Distance [112]) for text matching, the test bench simulates textual decay by modifying the annotated text. Ideally, this could be achieved by randomly switching characters using an automated script. However, given the relevance of keywords in annotations, as pointed out by Brush et al. [46], each annotation in the test bench was manually modified, in order to ensure that the modified annotations still represent English words. In most cases, the language grammar of modified annotated texts were also verified.

Building on the concept of the edit-distance algorithm [112], which detects changes to textual content as addition (i.e., addition of new characters), subtraction (i.e., removal of old characters), and replacement (i.e., addition of a new character for each removed character), the textual changes made to the annotations are also represented using a similar vocabulary. Textual decay is categorized into three primary types: *addition*, *removal*, and *replacement*. Each of these types are further divided into two types, based on the size of

ORIGINAL		TEXTUAL VARIATIONS	
		Annotation	Length
		This is a random annotation.	Length: 28
ADDITION	Less than 50%	This is a random textual annotation.	Length: 36
		This is a random textual annotation.	Variation: +28.5%
	More than 50%	This is a bit longer random textual annotation.	Length: 47
		This is a bit longer random textual annotation.	Variation: +67.8%
REMOVAL	Less than 50%	This is annotation.	Length: 19
		This is a random annotation.	Variation: -32.1%
	More than 50%	This anno.	Length: 10
		This is a random annotation.	Variation: -64.2%
REPLACEMENT	Less than 50%	This is some annotated text.	Length: 28
		This is a rand some annotated text .	Variation: 32.1%
	More than 50%	You see some annotated text.	Length: 28
		This is a rand You see some annotated text .	Variation: 60.7%

Figure 25 – Illustration of the different kinds of (textual) web page decay simulated in the test bench.

the change, i.e., *more than 50%* length of the annotated text versus *less than 50%* of the annotated text (see Figure 25). Combined, these different types of textual variances allow for simulation of six different types of textual decays.

6.2.2.2 Structural Decay Types This decay is primarily of two types. To reduce the complexity of the variance creation process, we only made minor structural changes to all annotated elements (see Fig. 26). As illustrated in Fig. 26, elements containing the annotated texts were either moved around as a single element, or were broken down into individual sub elements, and then moved around the document’s DOM. Moreover, we found that severe changes to both the text and structure of an annotation tends to render the annotation meaningless; therefore, for simulating structural decay, we only made structural changes to the original annotations, and to the three types (addition, removal, and replacement) of less than 50% textually changed annotations (discussed in Chapter 6.2.2.1). In total, we simulated four kinds of structural decays.

Together with the six kinds of textual changes and four kind of structural changes, the test bench is able to provide a total of 1200 unique web pages, containing 11,100 (textual and structural) variations. As the test bench is openly accessible on GitHub [143], to replicate the annotations and their variances, one only has to download the HTML files provided in the test bench’s repository. The repository also includes a step-by-step guide

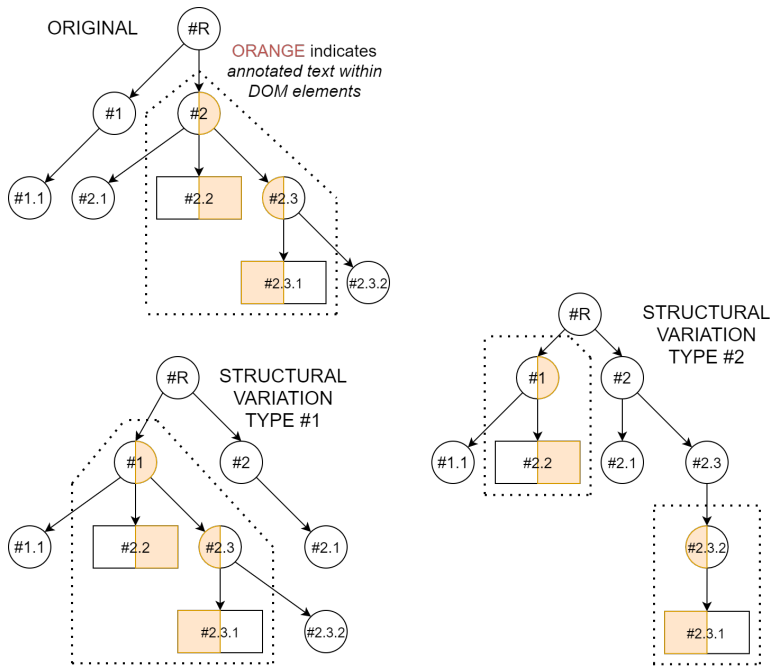


Figure 26 - Illustration of the different kinds of (structural) web page decay simulated in the test bench.

for setting up the test bench, and for evaluating other anchoring algorithms. A detailed view of the lengths of the original 1,110 annotations, and their corresponding textually decayed annotations (generated based on the methodology defined in Chapter 6.2.2.1) created for the test bench are presented in Table 8.

Figure 27 presents the distribution curve of the original annotation lengths and their density within the test bench. And Figures 28, 29, and 30 present the distribution curves of the modified annotation lengths (for the different textual decay types), their variances and their corresponding densities.

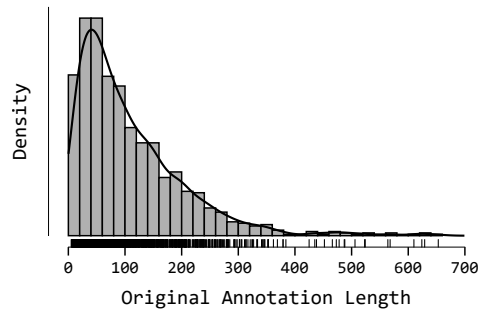


Figure 27 - Graph illustrating the distribution of the 1,110 original annotations (created for the test bench) with respect to their lengths.

Table 8 – Detailed view of the length of annotations and their corresponding textually decayed forms that are included in the test bench.

Annotation	Mode	Median	Mean	Range	Minimum	Maximum
Original Length	13.00	82.00	108.90	648.00	5.00	653.00
Text Added - Less Than 50%						
Length	17.00	105.00	136.18	810.00	6.00	816.00
Variance	20.00	28.00	28.19	42.00	7.00	49.00
Text Added - More Than 50%						
Length	43.00	137.00	184.32	1092.00	8.00	1100.00
Variance	62.00	68.00	70.25	45.00	53.00	98.00
Text Removed - Less Than 50%						
Length	10.00	63.00	84.62	574.00	4.00	578.00
Variance	20.00	23.00	23.28	43.00	5.00	48.00
Text Removed - More Than 50%						
Length	6.00	33.00	43.97	305.00	2.00	307.00
Variance	58.00	59.00	59.92	37.00	51.00	88.00
Text Replaced - Less Than 50%						
Length	13.00	82.00	108.90	648.00	5.00	653.00
Variance	25.00	25.00	25.39	39.00	10.00	49.00
Text Replaced - More Than 50%						
Length	13.00	82.00	108.90	648.00	5.00	653.00
Variance	55.00	60.00	61.66	40.00	52.00	92.00

Note. Total Valid Annotation Count = 1110

6.3 Results and Discussion

After designing the six types of textual decays and four types of structural decays, we integrated the decayed web pages into the test bench, and saved them into separate folders based on the decay types. We then visited each of the 1200 web pages, sequentially and viewed them using the Hypothes.is and Tippanee web annotation tools. By viewing the annotation created on these web pages through Hypothes.is and Tippanee, we were able to create a complete list of annotations, their corresponding decayed forms, and the state of the annotations. The states of each of the annotation viewed using Hypothes.is and Tippanee was categorized as: 'Y' indicating that the annotation was correctly attached, 'I' indicating that the annotation was incorrectly attached, and 'N' indicating that the annotation was orphaned. The number of annotations based on their states (i.e., Y, I, and N) and the various decay types, as viewed on Hypothes.is and Tippanee are enumerated in Tables 9 and 10, respectively. The tables also provide insights into the likelihood with which an annotation can be reattached using Hypothes.is' and Tippanee's anchoring algorithm, given the kind and veracity of a web page's decay.

To summarize, by comparing the 1,110 annotations and their 11,100 variances (included in proposed test bench) using the Hypothes.is and Tippanee web annotation systems, we find the following interesting results:

1. By designing this web annotation test bench, and testing it with the state of the art in web annotation systems (i.e., Hypothes.is) we have demonstrated that the robustness and accuracy of anchoring algorithms can be evaluated without relying on active web pages to decay.

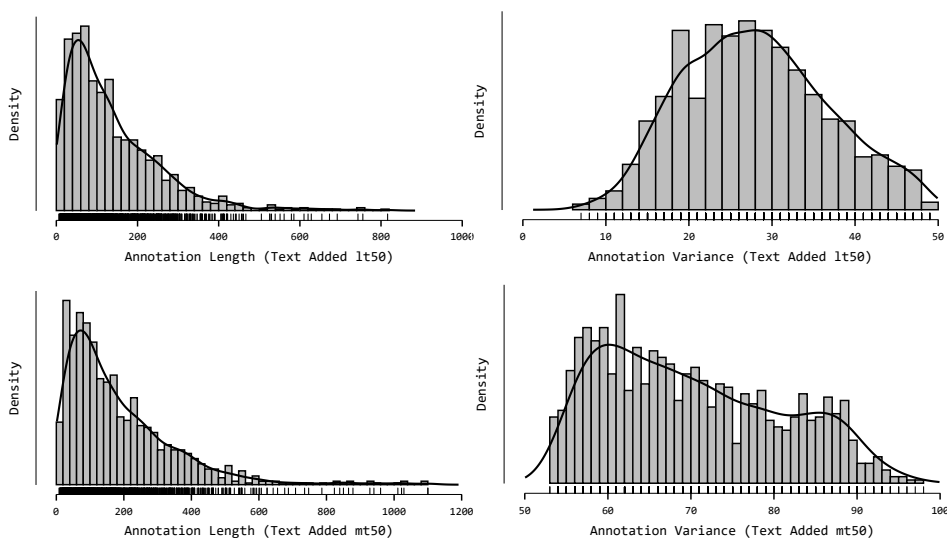


Figure 28 – Graph illustrating the distribution of the modified (i.e., text added) annotations (created for the test bench) with respect to their lengths and variances. (Top-Left) Distribution of annotation (with less than 50% text added) lengths versus number of annotations. (Top-Right) Distribution of annotation (with less than 50% text added) variances versus number of annotations. (Bottom-Left) Distribution of annotation (with more than 50% text added) lengths versus number of annotations. (Bottom-Right) Distribution of annotation (with more than 50% text added) variances versus number of annotations.

2. Among the ten different kinds of web page decay simulated in the test bench, Hypothes.is' fuzzy anchoring preformed better in only three types of decays, namely, when the annotated pages had structural change with no textual decay, when there was no structural change and only less than 50% text was added to the annotated content, and when there was no structural change and only less than 50% text of the annotated content was removed. For all remaining types of decays, Tippanee's DOM-oriented edit-distance algorithm performed better.
3. On an average, Hypothes.is' fuzzy anchoring algorithm attached 16.7% annotations to incorrect locations, while Tippanee's algorithm only incorrectly reattached 4.2% annotations.
4. Decayed annotations were more likely to be reattached to incorrect locations when using Hypothes.is, whereas Tippanee was more likely to show such annotations as orphans. It should be noted however that, Tippanee's users can view annotations in their original states even when they are orphaned.
5. Out of the 1,110 annotations and their 11,100 variances, Hypothes.is' fuzzy anchoring algorithm was able to correctly reattach 61.4% annotations. While, 16.7% annotations were attached incorrectly and 21.8% annotations were orphaned.
6. Out of the 1,110 annotations and their 11,100 variances, Tippanee's DOM-oriented edit-distance algorithm was able to correctly reattach 70.9% annotations. While, 4.2% annotations were attached incorrectly and 24.9% annotations were orphaned.

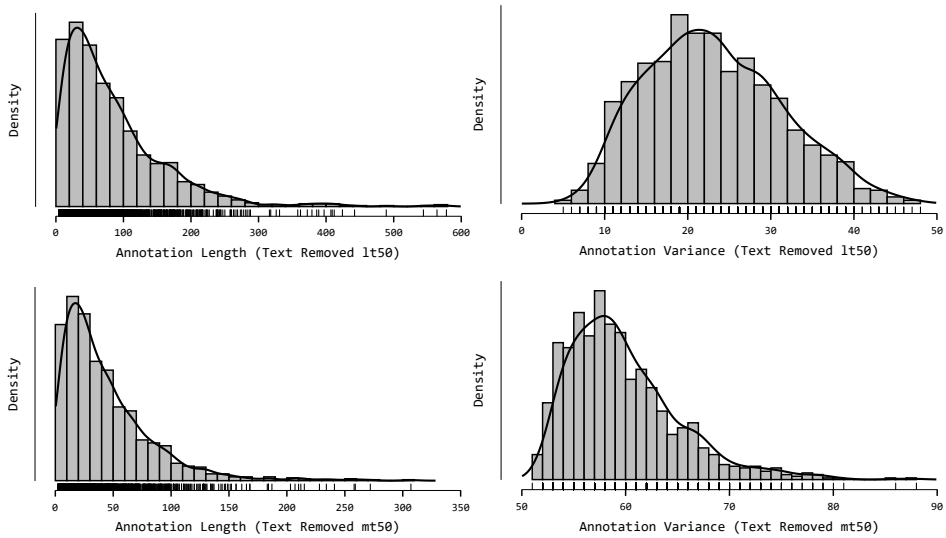


Figure 29 – Graph illustrating the distribution of the modified (i.e., text removed) annotations (created for the test bench) with respect to their lengths and variances. (Top-Left) Distribution of annotation (with less than 50% text removed) lengths versus number of annotations. (Top-Right) Distribution of annotation (with less than 50% text removed) variances versus number of annotations. (Bottom-Left) Distribution of annotation (with more than 50% text removed) lengths versus number of annotations. (Bottom-Right) Distribution of annotation (with more than 50% text removed) variances versus number of annotations.

7. Overall, Tippane’s anchoring algorithm was 9.5% more robust (i.e., it correctly reattached 9.5% more annotations) and 12.5% more accurate (i.e., it incorrectly reattached 12.5% less annotations) than that of Hypothes.is.
8. Figures 31, 32, 33 and 34, 35, 36 present flexplots [70] indicating the variances in annotation texts (after textual and structural decay) and their corresponding states (i.e., Y, I, and N) as seen on Hypothes.is and Tippane, respectively. The plots represent three decay conditions, namely, (i) where less than 50% of the annotate text is replaced, (ii) where less than 50% of the annotate text is replaced and the annotation is also structurally changed, and (iii) where more than 50% of the annotate text is replaced.

6.4 Summary and Conclusions

In this chapter, we addressed the challenge of evaluating the robustness of anchoring algorithms. To this end, we proposed a first of its kind test bench that enables the evaluation of anchoring algorithm’s robustness and accuracy. We first detailed the design choices that we were taken account during the development of the test bench. Based on insights from literature and from the experiments conducted as part of this work (see Chapters 4 and 5), we carefully accumulated 120 web pages from 50 popular websites. Using the Hypothes.is and Tippane web annotation tools, we then created 1,110 unique annotations on the accumulated web pages. After this, each of the annotated web pages was manually modified based on ten different web page decay types (including both textual and structural changes). In total, for the 1,110 annotations, we created 11,100 variations (each

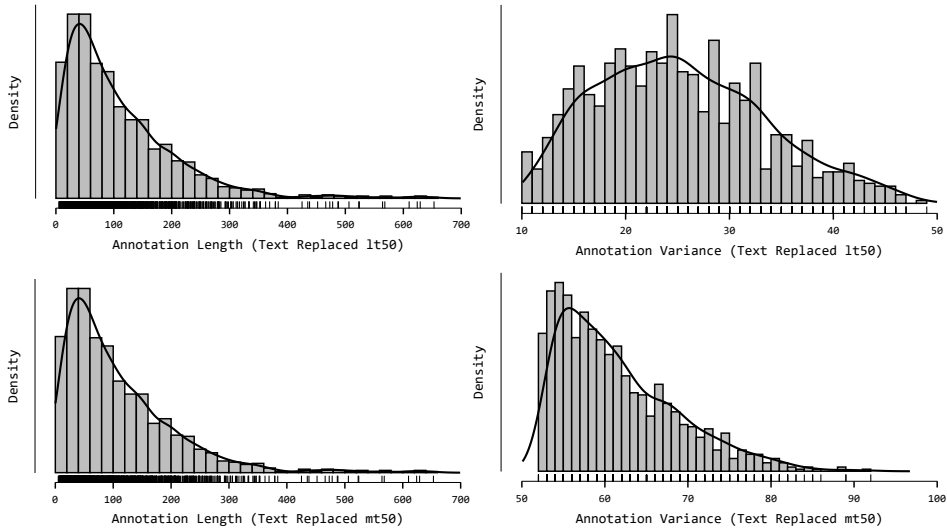


Figure 30 – Graph illustrating the distribution of the modified (i.e., text replaced) annotations (created for the test bench) with respect to their lengths and variances. (Top-Left) Distribution of annotation (with less than 50% text replaced) lengths versus number of annotations. (Top-Right) Distribution of annotation (with less than 50% text replaced) variances versus number of annotations. (Bottom-Left) Distribution of annotation (with more than 50% text replaced) lengths versus number of annotations. (Bottom-Right) Distribution of annotation (with more than 50% text replaced) variances versus number of annotations.

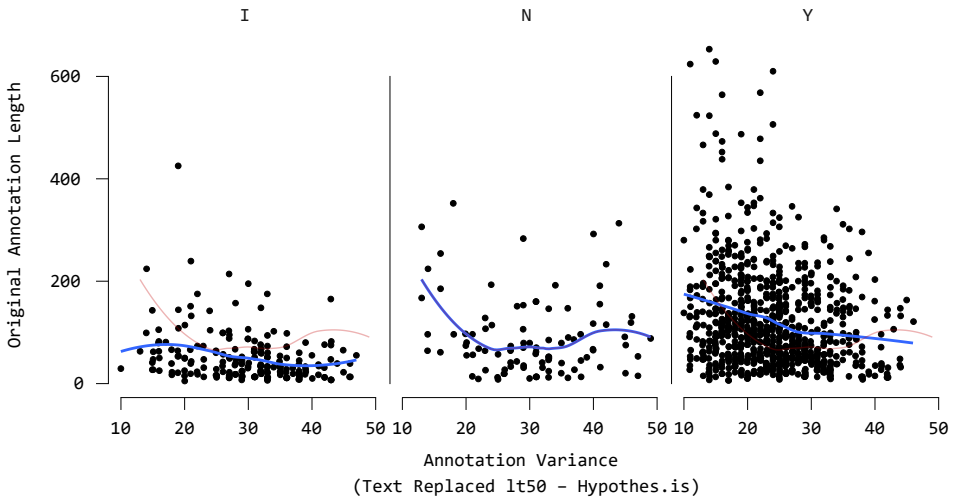


Figure 31 – Flexplot illustrating the variance in annotation texts, original annotation lengths, and their corresponding states (i.e., Y, I, and N). The plot illustrates the state of annotations as viewed on Hypothes.is, in the case where 1 - 49% of the annotated text was replaced.

simulating some form of decay).

In regards to the research question RQ2.3, the proposed web annotation test bench presented in this chapter demonstrates how the robustness of anchoring algorithms can

Table 9 – Detailed view of the states of annotations (indicated by Y, I, and N) corresponding to different decay types simulated in the test bench (when using Hypothes.is).

Level**	Total	Proportion	p	VS-MPR	Lower*	Upper*
Original Annotation with Structural Change						
I	70	0.06	2.07e-222	3.47e+218	0.05	0.08
N	148	0.13	1.18e-146	9.27e+142	0.11	0.15
Y	892	0.80	3.61e-97	4.59e+93	0.78	0.83
Text Added - Less Than 50%						
I	113	0.10	2.62e-177	3.45e+173	0.08	0.12
N	58	0.05	6.05e-237	1.12e+233	0.04	0.07
Y	939	0.85	7.92e-129	1.58e+125	0.82	0.87
Text Added - Less Than 50% with Structural Change						
I	130	0.12	7.66e-162	1.29e+158	0.10	0.14
N	237	0.21	5.60e-86	3.35e+82	0.19	0.24
Y	743	0.67	6.14e-30	8.90e+26	0.64	0.70
Text Added - More Than 50%						
I	187	0.17	2.26e-117	6.07e+113	0.15	0.19
N	68	0.06	9.22e-225	7.74e+220	0.05	0.08
Y	855	0.77	3.76e-76	5.64e+72	0.74	0.79
Text Removed - Less Than 50%						
I	116	0.10	1.71e-174	5.38e+170	0.09	0.12
N	60	0.05	1.89e-234	3.61e+230	0.04	0.07
Y	934	0.84	3.61e-125	3.55e+121	0.82	0.86
Text Removed - Less Than 50% with Structural Change						
I	136	0.12	1.19e-156	8.60e+152	0.10	0.14
N	220	0.20	5.98e-96	2.81e+92	0.18	0.22
Y	754	0.68	1.98e-33	2.47e+30	0.65	0.71
Text Removed - More Than 50%						
I	335	0.30	1.07e-40	3.75e+37	0.27	0.33
N	474	0.43	1.29e-6	20960.96	0.40	0.46
Y	301	0.27	3.30e-54	9.05e+50	0.25	0.30
Text Replaced - Less Than 50%						
I	197	0.18	1.41e-110	1.04e+107	0.16	0.20
N	96	0.09	5.22e-194	1.58e+190	0.07	0.10
Y	817	0.74	1.03e-57	2.71e+54	0.71	0.76
Text Replaced - Less Than 50% with Structural Change						
I	211	0.19	1.63e-101	9.71e+97	0.17	0.21
N	383	0.35	3.08e-25	2.11e+22	0.32	0.37
Y	516	0.46	0.02	4.57	0.44	0.49
Text Replaced - More Than 50%						
I	364	0.33	7.20e-31	7.36e+27	0.30	0.36
N	679	0.61	9.75e-14	1.26e+11	0.58	0.64
Y	67	0.06	6.00e-226	1.18e+222	0.05	0.08

Note. Total Valid Annotation Count = 1110

Also. Proportions tested against value: 0.5

** Levels - I: Incorrect | N: Orphan | Y: Correct

* 95% Confidence Interval for Proportion

Table 10 – Detailed view of the states of annotations (indicated by Y, I, and N) corresponding to different decay types simulated in the test bench (when using Tippanee).

Level**	Total	Proportion	p	VS-MPR	Lower*	Upper*
Original Annotation with Structural Change						
I	22	0.02	4.50e-286	1.25e+282	0.01	0.03
N	336	0.31	4.90e-39	8.51e+35	0.28	0.33
Y	743	0.67	1.36e-31	3.81e+28	0.65	0.70
Text Added - Less Than 50%						
I	39	0.04	8.08e-260	7.63e+255	0.03	0.05
N	13	0.01	3.89e-302	1.36e+298	6.30e-3	0.02
Y	1049	0.95	4.21e-242	1.57e+238	0.94	0.96
Text Added - Less Than 50% with Structural Change						
I	25	0.02	4.09e-281	1.39e+277	0.01	0.03
N	441	0.40	4.36e-11	3.54e+8	0.37	0.43
Y	635	0.58	3.92e-7	63556.62	0.55	0.61
Text Added - More Than 50%						
I	57	0.05	1.06e-235	6.42e+231	0.04	0.07
N	72	0.07	1.23e-217	6.00e+213	0.05	0.08
Y	972	0.88	1.70e-160	5.87e+156	0.86	0.90
Text Removed - Less Than 50%						
I	41	0.04	5.56e-257	1.12e+253	0.03	0.05
N	19	0.02	3.28e-291	1.68e+287	0.01	0.03
Y	1041	0.95	5.87e-232	1.18e+228	0.93	0.96
Text Removed - Less Than 50% with Structural Change						
I	30	0.03	3.44e-273	1.71e+269	0.02	0.04
N	379	0.34	2.69e-25	2.41e+22	0.32	0.37
Y	692	0.63	1.28e-17	7.40e+14	0.60	0.66
Text Removed - More Than 50%						
I	65	0.06	7.21e-226	9.84e+221	0.05	0.07
N	335	0.30	2.14e-39	1.93e+36	0.28	0.33
Y	701	0.64	9.20e-20	9.12e+16	0.61	0.67
Text Replaced - Less Than 50%						
I	54	0.05	1.62e-239	4.13e+235	0.04	0.06
N	104	0.09	1.16e-183	7.51e+179	0.08	0.11
Y	943	0.86	1.37e-136	8.58e+132	0.83	0.88
Text Replaced - Less Than 50% with Structural Change						
I	36	0.03	3.67e-264	1.65e+260	0.02	0.04
N	551	0.50	1.00	1.00	0.47	0.53
Y	514	0.47	0.03	3.50	0.44	0.50
Text Replaced - More Than 50%						
I	83	0.08	2.51e-205	3.12e+201	0.06	0.09
N	503	0.46	4.59e-3	14.88	0.43	0.49
Y	515	0.47	0.03	3.15	0.44	0.50

Note. Total Valid Annotation Count = 1101

Also. Proportions tested against value: 0.5

** Levels - I: Incorrect | N: Orphan | Y: Correct

* 95% Confidence Interval for Proportion

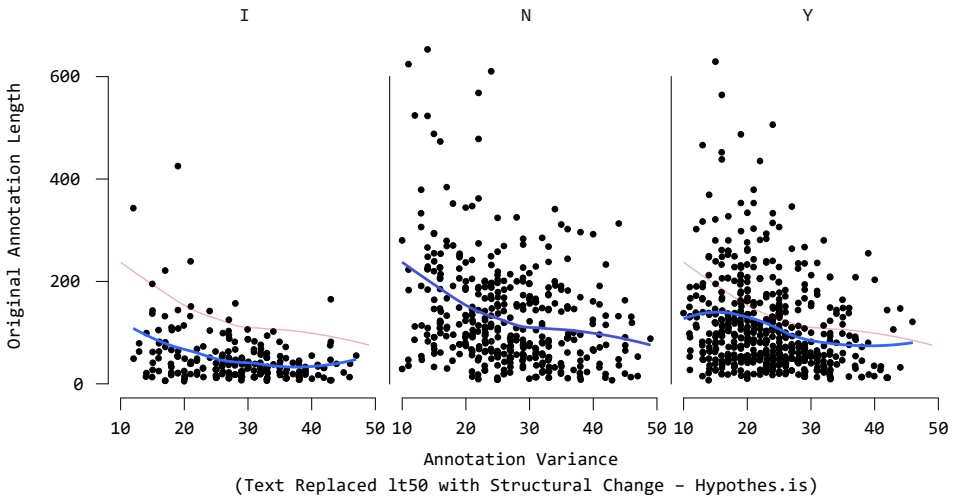


Figure 32 - Flexplot illustrating the variance in annotation texts, original annotation lengths, and their corresponding states (i.e., Y, I, and N). The plot illustrates the state of annotations as viewed on Hypothes.is, in the case where the annotated content was structurally changed and 1 - 49% of the annotated text was replaced.

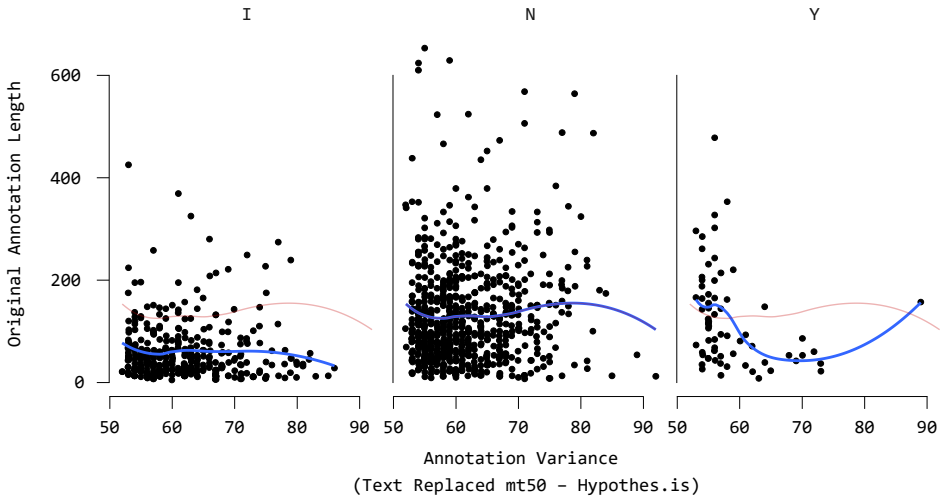


Figure 33 - Flexplot illustrating the variance in annotation texts, original annotation lengths, and their corresponding states (i.e., Y, I, and N). The plot illustrates the state of annotations as viewed on Hypothes.is, in the case where more than 50% of the annotated text was replaced.

be evaluated by simulating decay in web pages captured from the web. The test bench proposed in chapter contributes to the body of knowledge on web annotation systems and anchoring algorithm, and provides an alternative to the conventional methodology of creating annotations and observing them decay over time.

Furthermore, by comparing Hypothes.is' fuzzy anchoring algorithm with Tippanee's DOM-oriented edit-distance approach (i.e., dissertation contribution C2) using the pro-

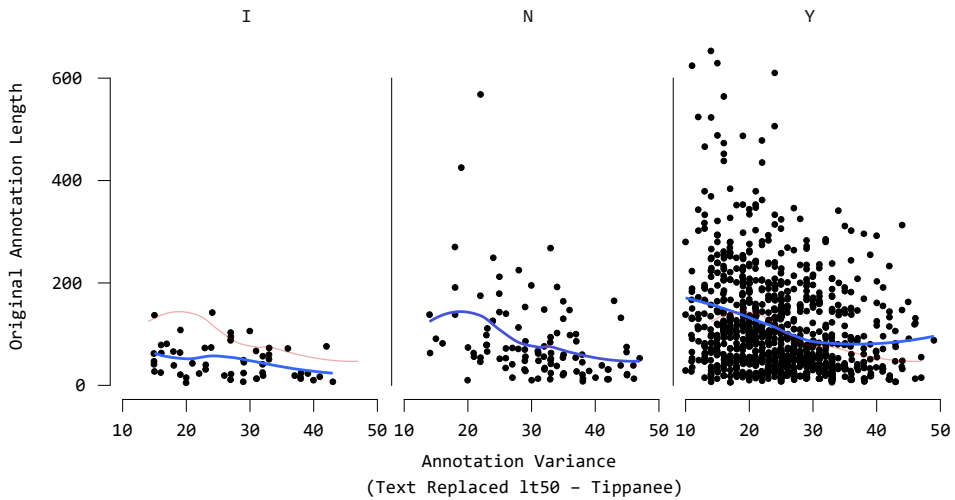


Figure 34 – Flexplot illustrating the variance in annotation texts, original annotation lengths, and their corresponding states (i.e., Y, I, and N). The plot illustrates the state of annotations as viewed on Tippianee, in the case where 1 - 49% of the annotated text was replaced.

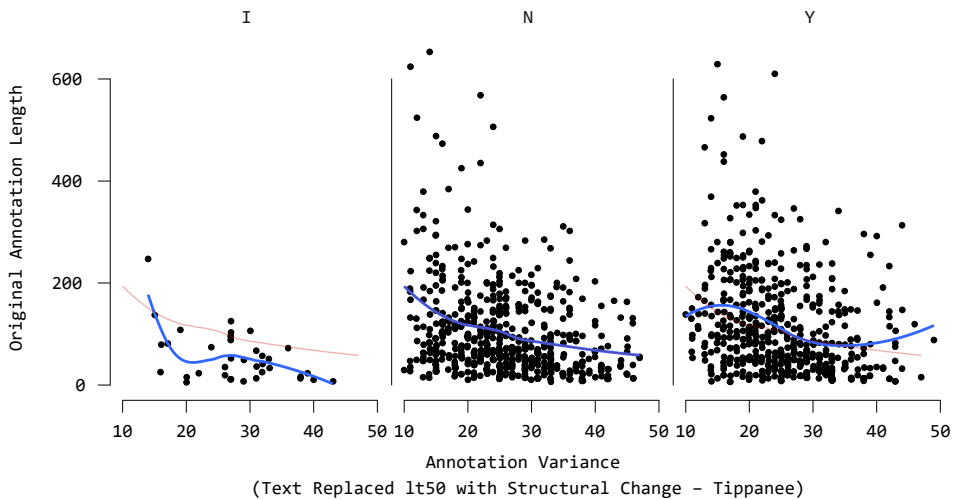


Figure 35 – Flexplot illustrating the variance in annotation texts, original annotation lengths, and their corresponding states (i.e., Y, I, and N). The plot illustrates the state of annotations as viewed on Tippianee, in the case where the annotated content was structurally changed and 1 - 49% of the annotated text was replaced.

posed test bench we are able to empirically validate the claim that the anchoring algorithm proposed in this dissertation is more robust and accurate than the current state-of-the-art. As illustrated in the chapter, Tippianee’s DOM-oriented edit-distance approach was found to be able to accurately reattach 9.5% more annotations than Hypothes.is; further supporting our initial finding from Chapter 4 that the algorithm proposed in this dissertation is more robust than Hypothes.is’ fuzzy anchoring algorithm.

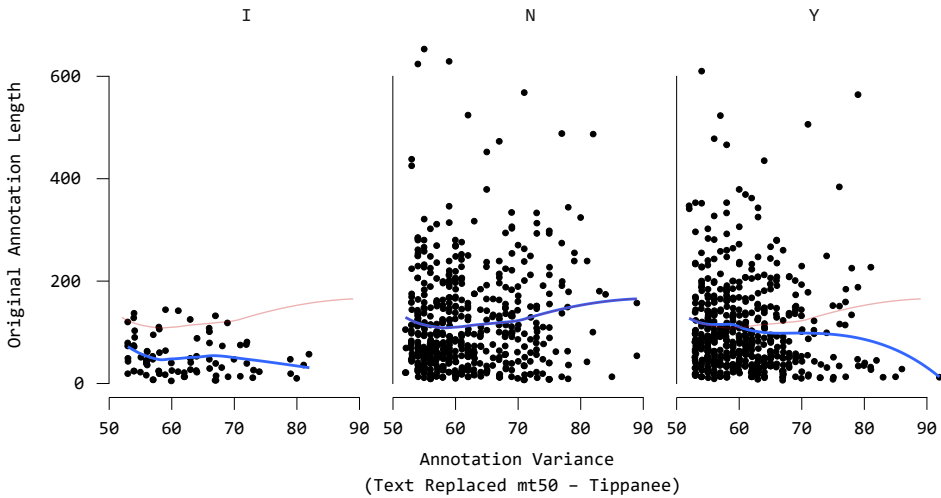


Figure 36 – Flexplot illustrating the variance in annotation texts, original annotation lengths, and their corresponding states (i.e., Y, I, and N). The plot illustrates the state of annotations as viewed on Tippane, in the case where more than 50% of the annotated text was replaced.

To refer back to the research question RQ2.1 raised in Chapter 4, based on the results presented in this chapter, it can be concluded that by including the structure and context of annotations during the anchor generation process (as demonstrated by the DOM-oriented edit-distance algorithm) annotations can be better prevented from being orphaned. We also found that storing the structure and context of annotations can be particularly useful, as it prevent annotations from being reattached at incorrect locations. Also, the additional information enables web annotation system users to view orphaned annotations and their context, thus enabling archival of annotated content at a user-level.

7 Conclusions

In this chapter, we provide an overview of the contributions presented in this dissertation. Following the discussion of our findings, a brief summary about the limitations of the presented work is provided. Finally, possible future directions that have been explored as part of this work are discussed.

7.1 Summary of Thesis Contributions

Web annotations have been an integral part of the World Wide Web, ever since early 1990s. This is evident from the numerous web annotation systems that were developed during the early years of the web. Although the use of web annotations systems in knowledge sharing and learning, is well studied in literature, researchers have found that alternative tools (based on Web 2.0 technologies) such as blogs, wikis, and discussion forums are more useful in learning. Furthermore, web annotation systems that do have support from academic and research communities, are unable to keep up with the ephemerality of web content. This has created a gap wherein web annotation systems either lack the features required to support knowledge creation and transfer (particularly in collaborative environments), or the underlying algorithms of these systems are unable to tackle the transient nature of web content, thereby rendering these systems ineffective. The work presented in this dissertation, aims to tackle both of these gaps by addressing two primary research questions (each with three sub-research questions):

RQ1 How to support knowledge creation and sharing through use of web annotations?

RQ1.1 What insights from knowledge management can be incorporated into web annotation processes?

RQ1.2 How can web annotations support knowledge management life cycles?

RQ1.3 What system features have to be fulfilled by solutions to support knowledge creation and sharing, when using web annotations?

RQ2 How to make textual web annotations robust against ephemerality of web content?

RQ2.1 How can web annotations be prevented from being orphaned?

RQ2.2 How to design a stable web annotation platform that does not fully rely on content providers?

RQ2.3 How to evaluate the robustness of anchoring algorithms?

The dissertation describes four distinct contributions. The first contribution is a framework that address the research question RQ1. While the remaining three contributions are a novel algorithm, and two web-based tools. Together these contributions address the research question RQ2.

The first contribution C1, is a set of web annotation activities (combined into a framework) that enable knowledge co-creation through use of socialization, externalization, combination and internalization. Developed by combining insights from literature on knowledge transformation with the well-established SECI model for knowledge creating companies, the proposed framework provides a set of textual and semantic annotation activities, for both online communities and organizations. It is argued that, by integrating the proposed activities and processes into a web annotation system, the system should be able to support collaborative knowledge creation and exchange. Many of the annotation activities proposed in the framework are integrated as system features into the novel web annotation tool presented as part of contribution C3.

The second contribution C2, is a novel anchoring algorithm that is more robust against ephemeral web content, in comparison to the current state-of-the-art. The algorithm utilizes three sets of information, namely the annotated text, its surrounding context (i.e., in textual form) and its structure. By combining these pieces of information, the algorithm is able to track annotated texts even when the underlying web page has decayed. The approach used by the proposed algorithm is distinct from any of the state-of-the-art anchoring algorithms, as it utilizes the already used fuzzy-string matching approach in combination with a novel fuzzy-DOM matching approach. By matching the DOM attributes of the annotated element, the algorithm is able to identify the correct source of the annotation even when text similar to the annotation appears multiple times on the same web page. The robustness of the proposed algorithm was evaluated by observing 735 annotations after a duration of one month. At the end of the experiment, the proposed algorithm was found to be 18.71% more robust than (the state-of-the-art) Hypothes.is' anchoring algorithm.

The third contribution C3 is a novel web annotation system, built around the novel anchoring algorithm. The aim behind developing the proposed web annotation system was twofold. First, in order to evaluate the proposed anchoring algorithm (i.e., C2) it was imperative to integrate the algorithm into a system so that it could be compared with the state-of-the-art anchoring algorithm. And second, to gauge whether the proposed novel web annotation activities (i.e., C1) could be integrated into a real-world web annotation system. Developed as a Google Chrome browser extension, the proposed web annotation system introduces several new system features that are currently not available in any state-of-the-art web annotation systems. One such key feature is that the system enables its users to store and reattach web annotation without the need for connecting to a server. Also, the system empowers its users by allowing them to archive pieces of information as web annotations. To evaluate the proposed system we conducted a lab experiment with participants, over a total duration of two weeks. At the end of the experiment we gathered user feedback, and found that most users found the annotation system useful and easy-to-use. We also concluded that most users preferred to use the system to annotate educational and news websites, Q&A portals, and search engines.

The fourth and final contribution C4 is a first of its kind test bench that enables its users to validate the robustness of anchoring algorithms. The proposed test bench is a critical contribution to the body of knowledge on web anchoring algorithms, as it provides a reproducible quantitative approach for evaluating anchoring algorithms. Composed by accumulating and creating 1,110 annotations and their 11,100 variations, the test bench simulates ten different forms of web page decay (including textual and structural changes). By comparing the states of annotations across the different decays, it becomes possible to empirically validate the robustness of anchoring algorithms. An additional reasoning behind the development of the test bench was to validate the robustness and accuracy of the novel anchoring algorithm (i.e., C2). Using the test bench, we were able to compare the novel DOM-oriented edit-distance algorithm with Hypothes.is' fuzzy anchoring algorithm. The novel algorithm was found to be 9.5% more robust and 12.5% more accurate than Hypothes.is' algorithm.

Together the contributions C2, C3, and C4, address the research question RQ2. By aggregating the findings from the experiments conducted for each of the contributions, it is concluded that textual web annotations can be made more robust against ephemerality of web content, by storing both textual and structural information about the annotated content into the anchor data. And that doing so, can improve both the accuracy and robustness of future anchoring algorithms.

While some of the various annotation activities proposed in this work (i.e., C1) were integrated into the novel web annotation system Tippanee (i.e., C3), most of these activities added as system features only support textual annotations. As pointed in the discussion in Chapter 3, adding and using semantic annotation features (and activities) requires further examination before technological solutions supporting the same can be developed. Also, it needs to be understood if and how (non-expert) web users would be able to make better use of the proposed semantic annotation activities. This is critical for the evaluation of the proposed framework (i.e., C1) in its totality. Hence, this is planned as future work. In regards to the proposed anchoring algorithm (i.e., C2), although after rigorous evaluation the algorithm was found to be more robust and accurate than the state-of-the-art, the algorithm could potentially be optimized future by improving its underlying parameters. For instance, as pointed out in Chapter 4.2.2, the similarity index threshold for deciding whether an annotation is orphaned was set to be 50%, however, as evident from the results presented in Chapter 6.3, this implies that proposed algorithm is more likely to return decayed annotations as orphans (specially when compared to Hypothes.is). By further optimizing the algorithm's parameters and re-evaluating it against its different versions (i.e., with different parameters), the algorithm's accuracy and robustness could be improved further in the future.

7.2 Future Directions

This section presents some broader questions and potential next steps for further empowering the users of web annotation systems. The artifacts (i.e., framework, algorithm, and tools) presented in the dissertation, are only the first steps towards building better web annotation systems, are therefore there is much more that needs to be explored, examined and improved, before web annotation systems are able to achieve their full potential.

7.2.1 Adding a Layer of Security Through Peer-to-Peer

One of the key consideration taken in account while designing the present version of the web annotation system Tippanee was to empower web users, by enabling them to annotate, store and reattach web annotations without needing them to connect to a server to access (and use) said services. This was achieved by allowing users to store their annotations on their personal (local) machines. However, if users wanted to share the annotations with other system users, they were still required to connect to a server. This implied that if users wanted to work in groups, they would still have to rely on the annotation service provider. The issue can be resolved by developing a solution that allows users to exchange information (i.e., in the present context, annotations) without requiring to connect to a server. One such solution that we explored was peer-to-peer (P2P) message transfer. By developing a P2P-based chat application for web browsers, in **publication III** and **publication IV**, we successfully demonstrated that users can collaboratively exchange messages directly through web browsers without the need for relying on a server. We also demonstrated that by enabling P2P message-exchange over browsers users' messages (and data) could be prevented from being stolen. By integrating this P2P-based message-exchange feature into web annotation systems, users of such systems could be empowered even further, giving them full control and access over their annotations at all time, even in collaborative environments.

7.2.2 Guidelines for Online Collaborative Communities

In **publication V**, we developed a novel framework to support the design and development of crowd-oriented ICT solutions. The insights from the framework were taken into account during the development of web annotation system Tippanee. However, some of the frameworks components such as user reputation, user hierarchy, and contests, could not be integrated into the web annotation system at the time. As evident from the findings of **publication VI**, system components such as user reputation and user hierarchy can play a critical role in organizational settings, as these components can support users (i.e., employees) in making better decisions, and can help in establishing trust among group members. This trust can also play a vital role in online web communities, and therefore, further investigation is required before the 'generic' CI framework [**publication V**] can be completely integrated into web annotation systems.

7.2.3 Annotations in Support of an Open, Democratic Web

Other components of the 'generic' CI framework [**publication V**], including contests, group decisions and motivations can also have critical impact on web annotation communities and the web at-large. As we eluded to at the beginning of the dissertation, the web today has started to "wane in the face of a *nasty storm* of issues" [170]. Many of the issues being encountered on the web, are actively being addressed using solutions that harness the wisdom of crowds (i.e., CI). Some of these solutions also utilize annotation systems as tools to tackle issues such fake news and misinformation. Users of such (specialized) systems annotate news article across multiple sources, which are then aggregated to verify the validity of the annotated information. These systems however, lack the feature sets that are inherent to conventional web annotation systems, thus preventing web users from using these specialized annotation systems on a large scale. We would argue that integrating the features of these (specialized) systems, and their CI-related concepts into conventional web annotation systems, could enable the development of novel tools that could motivate and empower web-users to more actively engage (i.e., curate and critique) with web content. Thus supporting in the development of a more open and democratic web.

List of Figures

1	Illustration of the W3C's WAD model.....	20
2	Illustration of decay of cited links with respect to time.	23
3	Illustration of Nonaka et al.'s SECI model.....	29
4	Illustration of social software features weaved into ERP systems.....	31
5	Illustration of SECI annotation activities	34
6	Illustration of proposed annotation framework	37
7	Illustration of HTML DOM.....	40
8	Illustration of annotated content and context as explained by Brush et al. ...	41
9	Architecture of Hypothes.is' anchoring sub-systems.	42
10	Illustration of Hypothes.is incorrectly reattaching annotation.	45
11	Flowchart illustrating the proposed DOM-Oriented Edit-Distance anchoring algorithm.....	48
12	Similarity indexes of 675 annotations studied during evaluation of proposed algorithm.	50
13	Illustration of Tiptanee User Interface.	55
14	System Architecture of Proposed Web Annotation Tool.....	55
15	Architecture of Tiptanee's anchoring subsystems.....	56
16	Illustration of Tiptanee's Similarity Index feature.	57
17	Annotation graphs as viewed on Tiptanee.....	58
18	Pie chart illustrating the distribution of participants' response to the question Q7.	60
19	Pie chart illustrating the distribution of participants' response to the question Q8.	60
20	Graph illustrating the relationship between participants' responses from questions Q7 and Q8.	61
21	Heat-map illustrating Pearson's correlation between participants' response to the question Q1 - Q6 and Q9.....	63
22	Graph illustrating growth in Tiptanee's total user count since 2018.....	64
23	Illustration of the composition of the proposed test bench.....	66
24	Illustration of the different annotation types included in the test bench.....	67
25	Illustration of the different kinds of (textual) web page decay simulated in the test bench.	69
26	Illustration of the different kinds of (structural) web page decay simulated in the test bench.....	70
27	Graph illustrating the distribution of the 1,110 original annotations (created for the test bench) with respect to their lengths.	70
28	Graph illustrating the distribution of the modified (i.e., text added) annotations (created for the test bench) with respect to their lengths and variances.	72
29	Graph illustrating the distribution of the modified (i.e., text removed) annotations (created for the test bench) with respect to their lengths and variances.	73
30	Graph illustrating the distribution of the modified (i.e., text replaced) annotations (created for the test bench) with respect to their lengths and variances.....	74
31	Flexplot illustrating the variance in annotation texts, original annotation lengths, and their corresponding states (i.e., Y, I, and N) - Part 1.....	74

32	Flexplot illustrating the variance in annotation texts, original annotation lengths, and their corresponding states (i.e., Y, I, and N) - Part 2.	77
33	Flexplot illustrating the variance in annotation texts, original annotation lengths, and their corresponding states (i.e., Y, I, and N) - Part 3.	77
34	Flexplot illustrating the variance in annotation texts, original annotation lengths, and their corresponding states (i.e., Y, I, and N) - Part 4.	78
35	Flexplot illustrating the variance in annotation texts, original annotation lengths, and their corresponding states (i.e., Y, I, and N) - Part 5.	78
36	Flexplot illustrating the variance in annotation texts, original annotation lengths, and their corresponding states (i.e., Y, I, and N) - Part 6.	79

List of Tables

1	Research questions with associated dissertation chapters and contributions.	15
2	Mapping of dissertation contributions, proposed artifacts and corresponding evaluation methodologies.	15
3	Summary of participants' response to the questions Q1 - Q6, and Q9.....	59
4	Summary of participants' response to the question Q7.....	60
5	Summary of participants' response to the question Q8.....	60
6	Contingency table mapping the relationship between participants' responses from questions Q7 and Q8.	61
7	Pearson's correlation between participants' response to the question Q1 - Q6 and Q9.	62
8	Detailed view of the length of annotations and their corresponding textually decayed forms that are included in the test bench.....	71
9	Detailed view of the states of annotations (indicated by Y, I, and N) corresponding to different decay types simulated in the test bench (when using Hypothes.is).	75
10	Detailed view of the states of annotations (indicated by Y, I, and N) corresponding to different decay types simulated in the test bench (when using Tippanee).	76

Listings

1	Example of HTML code from Wikipedia homepage.	40
2	Example of JSON object generated by the novel anchoring algorithm. . .	46
3	Example of match and mismatch probability generated by the novel anchoring algorithm, during annotation reattachment.	47

References

- [1] [accessed: 12th Oct. 2021] <https://github.com/openannotation/annotator>.
- [2] [accessed: 12th Oct. 2021] <http://a.nnotate.com/>.
- [3] [accessed: 12th Oct. 2021] <http://annotatorjs.org/>.
- [4] [accessed: 12th Oct. 2021] <https://genius.com/web-annotator>.
- [5] [accessed: 12th Oct. 2021] <https://opensource.google/projects/diff-match-patch>.
- [6] [accessed: 12th Oct. 2021] <https://perma.cc/>.
- [7] [accessed: 12th Oct. 2021] <https://web.hypothes.is/>.
- [8] [accessed: 12th Oct. 2021] <https://web.hypothes.is/blog/fuzzy-anchoring/>.
- [9] [accessed: 12th Oct. 2021] <https://web.hypothes.is/blog/our-view-from-20-million-annotations/>.
- [10] [accessed: 12th Oct. 2021] <https://web.hypothes.is/partners/>.
- [11] [accessed: 12th Oct. 2021] <https://www.diigo.com/>.
- [12] [accessed: 12th Oct. 2021] <https://www.w3.org/2001/Annotea/>.
- [13] [accessed: 12th Oct. 2021] <https://www.w3.org/TR/annotation-model/>.
- [14] [accessed: 12th Oct. 2021] <https://xanadu.com.au/ted/XU/XuPageKeio.html>.
- [15] [accessed: 12th Oct. 2021] <https://xanadu.com/zigzag/>.
- [16] [accessed: 12th Oct. 2021] <http://www.ncsa.illinois.edu/enabling/mosaic/>.
- [17] [accessed: 12th Oct. 2021] <http://zesty.ca/crit/>.
- [18] [accessed: 12th Oct. 2021] <http://zesty.ca/crit/ht98.html>.
- [19] The second international WWW conference '94: Mosaic and the web. National Center for Supercomputing Applications (NCSA), 1994.
<https://archive.org/details/www-conf-fall-1994>; Accessed: October 12, 2021.
- [20] Iso/iec/ieee international standard - systems and software engineering-vocabulary. *ISO/IEC/IEEE 24765:2017(E)*, pages 1–541, 2017.
- [21] M. Abe and M. Hori. A visual approach to authoring xpath expressions. *Markup Lang.*, 3(2):191–212, Apr. 2001.
- [22] P. Anderson. *What is Web 2.0? Ideas, technologies and implications for education*. JISC Bristol, 2007.
- [23] M. Andreessen. Group annotation server guinea pigs? The World Wide Web History Project, 1993.
<http://1997.webhistory.org/www.lists/www-talk.1993q2/0416.html>;
Accessed: October 12, 2021.

- [24] C. Asakawa and H. Takagi. Annotation-based transcoding for nonvisual web access. In *Proceedings of ASSETS'00: the 4th ACM SIGCAPH Conference on Assistive Technologies*, New York, NY, USA, 2000. ACM.
- [25] A. Atrash, M.-H. Abel, and C. Moulin. Notes and annotations as information resources in a social networking platform. *Computers in Human Behavior*, 51:1261–1267, 2015.
- [26] M. Aturban, M. L. Nelson, and M. C. Weigle. Quantifying Orphaned Annotations in Hypothes.is. In *Research and Advanced Technology for Digital Libraries: 19th International Conference on Theory and Practice of Digital Libraries, TPDL 2015, Poznań, Poland, September 14–18, 2015, Proceedings*, pages 15–27, Cham, 2015. Springer International Publishing.
- [27] F. Bagayogo, L. Lapointe, J. Ramaprasad, and I. Vedel. Co-creation of knowledge in healthcare: A study of social media usage. In *2014 47th Hawaii International Conference on System Sciences*. IEEE, Jan. 2014.
- [28] J. B. Bak-Coleman, M. Alfano, W. Barfuss, C. T. Bergstrom, M. A. Centeno, I. D. Couzin, J. F. Donges, M. Galesic, A. S. Gersick, J. Jacquet, A. B. Kao, R. E. Moran, P. Romanczuk, D. I. Rubenstein, K. J. Tombak, J. J. Van Bavel, and E. U. Weber. Stewardship of global collective behavior. *Proceedings of the National Academy of Sciences*, 118(27), 2021.
- [29] V. Bakir and A. McStay. Fake News and The Economy of Emotions. *Digital Journalism*, 6(2):154–175, 2018.
- [30] Z. Bar-Yossef, A. Z. Broder, R. Kumar, and A. Tomkins. Sic transit gloria telae: Towards an understanding of the web's decay. In *Proceedings of the 13th International Conference on World Wide Web, WWW '04*, page 328–337, New York, NY, USA, 2004. Association for Computing Machinery.
- [31] N. H. F. Beebe. Group annotations in NCSA Mosaic. University of Utah, 1995. <https://perma.cc/NU6T-F2E9>; Accessed: October 26, 2021.
- [32] M. Beno, E. Filtz, S. Kirrane, and A. Polleres. Doc2rdfa: Semantic annotation for web documents. In M. Alam, R. Usbeck, T. Pellegrini, H. Sack, and Y. Sure-Vetter, editors, *Proceedings of the Posters and Demo Track of the 15th International Conference on Semantic Systems co-located with 15th International Conference on Semantic Systems (SEMANTICS 2019), Karlsruhe, Germany, September 9th - to - 12th, 2019*, volume 2451 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.
- [33] C. Bereiter. *Education and Mind in the Knowledge Age*. Routledge, Apr. 2002.
- [34] H. Berghel. Malice domestic: The cambridge analytica dystopia. *Computer*, 51(5):84–89, 2018.
- [35] T. Berners-Lee. History of the web. World Wide Web Foundation. <https://webfoundation.org/about/vision/history-of-the-web/>; Accessed: October 12, 2021.
- [36] T. Berners-Lee. Information management: A proposal. Technical report, CERN, 1989.

- [37] T. Berners-Lee. Www: past, present, and future. *Computer*, 29(10):69–77, 1996.
- [38] T. Berners-Lee. Long live the web. *Scientific American*, 303(6):80–85, 2010.
- [39] T. Berners-Lee and M. Fischetti. *Weaving the web: The original design and ultimate destiny of the World Wide Web by its inventor*. Harper Collins Publishers, 1999.
- [40] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.
- [41] C. Bizer, T. Heath, and T. Berners-Lee. Linked data. In *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pages 205–227. IGI Global, 2011.
- [42] C. Bizer, T. Heath, K. Idehen, and T. Berners-Lee. Linked data on the web (Ildow2008). In *Proceedings of the 17th International Conference on World Wide Web, WWW '08*, page 1265–1266, New York, NY, USA, 2008. Association for Computing Machinery.
- [43] M. Bonn and J. McGlone. New feature: Article annotation with hypothes.is. *The Journal of Electronic Publishing*, 17(2), May 2014.
- [44] T. Brants and O. Plaehn. Interactive corpus annotation. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece, May 2000. European Language Resources Association (ELRA).
- [45] A. J. Brush and D. Bargeron. Robustly Anchoring Annotations Using Keywords. Technical report, Microsoft, Nov 2001.
- [46] A. J. B. Brush, D. Bargeron, A. Gupta, and J. J. Cadiz. Robust annotation positioning in digital documents. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '01*, page 285–292, New York, NY, USA, 2001. Association for Computing Machinery.
- [47] H. Byrne. Reviewing football history through the uk web archive. *Soccer & Society*, 21(4):461–474, 2020.
- [48] G. Caldarelli, R. D. Nicola, F. D. Vigna, M. Petrocchi, and F. Saracco. The role of bot squads in the political propaganda on twitter. *Communications Physics*, 3(1), May 2020.
- [49] P. R. Carlile. A pragmatic view of knowledge and boundaries: Boundary objects in new product development. *Organization Science*, 13(4):442–455, 2002.
- [50] P. R. Carlile. Transferring, translating, and transforming: An integrative framework for managing knowledge across boundaries. *Organization Science*, 15(5):555–568, 2004.
- [51] J. W. Chan and J. W. Pow. The role of social annotation in facilitating collaborative inquiry-based learning. *Computers & Education*, 147:103787, 2020.
- [52] X. Chen, S. Jia, and Y. Xiang. A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications*, 141:112948, 2020.
- [53] P. Ciccarese, S. Soiland-Reyes, and T. Clark. Web Annotation as a First-Class Object. *IEEE Internet Computing*, 17(6):71–75, Nov 2013.

- [54] M. Cinelli, G. De Francisci Morales, A. Galeazzi, W. Quattrociocchi, and M. Starnini. The echo chamber effect on social media. *Proceedings of the National Academy of Sciences*, 118(9), 2021.
- [55] M. Costa, D. Gomes, and M. J. Silva. The evolution of web archiving. *International Journal on Digital Libraries*, 18(3):191–205, Sep 2017.
- [56] H. Cunningham. Gate, a general architecture for text engineering. *Computers and the Humanities*, 36(2):223–254, 2002.
- [57] J. R. Davis and D. P. Huttenlocher. Shared annotation for cooperative learning. In *The First International Conference on Computer Support for Collaborative Learning*, CSCS '95, page 84–88, USA, 1995. L. Erlbaum Associates Inc.
- [58] L. Denoue and L. Vignollet. An annotation tool for web browsers and its applications to information retrieval. In *Content-Based Multimedia Information Access - Volume 1*, RIAO '00, page 180–195, Paris, FRA, 2000. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE.
- [59] J. v. Dijck. *The culture of connectivity*. Oxford University Press, 2013.
- [60] D. Draheim. On the radical de- and re-construction of today's enterprise applications. In *Proceedings of CENTERIS'2019 - 10th Conference on Enterprise Information Systems*. Elsevier, forthcoming in 2019.
- [61] D. Draheim, M. Felderer, and V. Pekar. Weaving social software features into enterprise resource planning systems. In F. Piazzolo and M. Felderer, editors, *Novel Methods and Technologies for Enterprise Information Systems*, pages 223–237, Cham, 2014. Springer International Publishing.
- [62] J. S. Dryzek, A. Bächtiger, S. Chambers, J. Cohen, J. N. Druckman, A. Felicetti, J. S. Fishkin, D. M. Farrell, A. Fung, A. Gutmann, H. Landmore, J. Mansbridge, S. Marien, M. A. Neblo, S. Niemeyer, M. Setälä, R. Slothuus, J. Suiter, D. Thompson, and M. E. Warren. The crisis of democracy and the science of deliberation. *Science*, 363(6432):1144–1146, 2019.
- [63] V. Dwivedi, V. Pattanaik, V. Deval, A. Dixit, A. Norta, and D. Draheim. Legally enforceable smart-contract languages: A systematic literature review. *ACM Comput. Surv.*, 54(5), June 2021.
- [64] I. Dylko, I. Dolgov, W. Hoffman, N. Eckhart, M. Molina, and O. Aaziz. The dark side of technology: An experimental investigation of the influence of customizability technology on online political selective exposure. *Computers in Human Behavior*, 73:181–190, Aug 2017.
- [65] L. Ehrlinger and W. WöB. Towards a definition of knowledge graphs. In *SEMANTICS (Posters, Demos, SuCCESS)*, 2016.
- [66] B. Eric. Decisions 2.0: the Power of Collective Intelligence. *MIT Sloan Management Review*, 50(2):45, 2009.
- [67] S. Faraj and M. M. Wasko. The web of knowledge: An investigation of knowledge exchange in networks of practice. CiteSeerX, 2001. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.4559>; Accessed: October 12, 2021.

- [68] Z. T. Fernando, I. Marenzi, and W. Nejdl. ArchiveWeb: collaboratively extending and exploring web archive collections—how would you like to work with your collections? *International Journal on Digital Libraries*, 19(1):39–55, Jan. 2017.
- [69] Z. T. Fernando, I. Marenzi, W. Nejdl, and R. Kalyani. Archiveweb: Collaboratively extending and exploring web archive collections. In N. Fuhr, L. Kovács, T. Risse, and W. Nejdl, editors, *Research and Advanced Technology for Digital Libraries*, pages 107–118, Cham, 2016. Springer International Publishing.
- [70] D. Fife. Flexplot: Graphically-based data analysis. *Psychological Methods*, Nov. 2021.
- [71] T. Filipowski, P. Kazienko, P. Bródka, and T. Kajdanowicz. Web-based knowledge exchange through social links in the workplace. *Behaviour & Information Technology*, 31(8):779–790, Aug. 2012.
- [72] K. Fontichiaro and J. A. Oehrli. Why data literacy matters. *Knowledge quest*, 44(5):21–27, 2016.
- [73] Frank Kappe, Hermann Maurer, and Keith Andrews. The hyper-g network information system, 1995.
- [74] S. Fu, G.-J. de Vreede, X. Cheng, I. Seeber, R. Maier, and B. Weber. Convergence of Crowdsourcing Ideas : A Cognitive Load Perspective. In *Thirty eighth International Conference on Information Systems*, number 2 in ICIS '17, pages 1–11. Association for Information Systems, 2017.
- [75] I. Glover, Z. Xu, and G. Hardaker. Online annotation – research and practices. *Computers & Education*, 49(4):1308–1320, 2007.
- [76] D. Gomes, S. Freitas, and M. J. Silva. Design and selection criteria for a national web archive. In J. Gonzalo, C. Thanos, M. F. Verdejo, and R. C. Carrasco, editors, *Research and Advanced Technology for Digital Libraries*, pages 196–207, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [77] Y. Gorodnichenko, T. Pham, and O. Talavera. Social media, sentiment and public opinions: Evidence from #brexit and #uselection. *European Economic Review*, 136:103772, 2021.
- [78] A. M. Guess, M. Lerner, B. Lyons, J. M. Montgomery, B. Nyhan, J. Reifler, and N. Sircar. A digital media literacy intervention increases discernment between mainstream and false news in the united states and india. *Proceedings of the National Academy of Sciences*, 117(27):15536–15545, 2020.
- [79] A. Gupta and V. Pattanaik. Survey paper on encryption, authentication and auditing services for better cloud security. *International Journal of Computer Applications*, 138(10):33–37, Mar. 2016.
- [80] A. Gupta, V. Pattanaik, and M. Singh. Enhancing k means by unsupervised learning using PSO algorithm. In *2017 International Conference on Computing, Communication and Automation (ICCCA)*. IEEE, May 2017.
- [81] J. Hemsley and R. M. Mason. Knowledge and knowledge management in the social media age. *Journal of Organizational Computing and Electronic Commerce*, 23(1-2):138–167, 2013.

- [82] J. Hendler. Web 3.0 emerging. *Computer*, 42(1):111–113, 2009.
- [83] A. R. Hevner, S. T. March, J. Park, and S. Ram. Design science in information systems research. *MIS Quarterly*, 28(1):75–105, 2004.
- [84] Hitkul, A. Prabhu, D. Guhathakurta, J. Jain, M. Subramanian, M. Reddy, S. Sehgal, T. Karandikar, A. Gulati, U. Arora, R. R. Shah, and P. Kumaraguru. Capitol (pat)riots: A comparative study of twitter and parler. arXiv.org, 2021.
<https://arxiv.org/abs/2101.06914>; Accessed: October 12, 2021.
- [85] A. Hogan, E. Blomqvist, M. Cochez, C. D’amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann. Knowledge graphs. *ACM Comput. Surv.*, 54(4), jul 2021.
- [86] M. Hori, G. Kondoh, K. Ono, S. Ichi Hirose, and S. Singhal. Annotation-based web content transcoding. *Computer Networks*, 33(1):197–211, 2000.
- [87] P. Iannuzzi, M. Eisenberg, D. W. Farmer, C. Gibson, L. A. Goetsch, B. Lessin, B. G. Lindauer, H. B. Rader, O. Ratteray, and A. H. Jenkins. ACRL STANDARDS: Information literacy competency standards for higher education. *American Library Association*, 61(3):207–215, Mar. 2000.
- [88] J. Introne, R. Laubacher, G. Olson, and T. Malone. Solving Wicked Social Problems with Socio-computational Systems. *KI - Künstliche Intelligenz*, 27(1):45–52, Feb 2013.
- [89] M. Jakubik. Practice Ecosystem of Knowledge Co-Creation. *International Journal of Management, Knowledge and Learning*, 7(2):199–216, 2018.
- [90] K. H. Jamieson and J. N. Cappella. *Echo chamber: Rush Limbaugh and the conservative media establishment*. Oxford University Press, 2008.
- [91] S. M. Jang and J. K. Kim. Third person effects of fake news: Fake news regulation and media literacy interventions. *Computers in Human Behavior*, 80:295–302, 2018.
- [92] L. Jasny, J. Waggle, and D. R. Fisher. An empirical examination of echo chambers in US climate policy networks. *Nature Climate Change*, 5(8):782–786, May 2015.
- [93] B. Jayles, C. Sire, and R. H. J. M. Kurvers. Impact of sharing full versus averaged social information on social influence and estimation accuracy. *Journal of The Royal Society Interface*, 18(180):20210231, 2021.
- [94] S. M. Jones-Jang, T. Mortensen, and J. Liu. Does media literacy help identification of fake news? information literacy helps, but other literacies don’t. *American Behavioral Scientist*, 65(2):371–388, 2021.
- [95] J. Jull, A. Giles, and I. D. Graham. Community-based participatory research and integrated knowledge translation: advancing the co-creation of knowledge. *Implementation Science*, 12(1), Dec. 2017.
- [96] A. J. Kalafut, M. Gupta, C. A. Cole, L. Chen, and N. E. Myers. An empirical study of orphan dns servers in the internet. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC ’10, page 308–314, New York, NY, USA, 2010. Association for Computing Machinery.

- [97] A. Kalboussi, O. Mazhoud, and A. H. Kacem. Comparative study of web annotation systems used by learners to enhance educational practices: features and services. *International Journal of Technology Enhanced Learning*, 8(2):129, 2016.
- [98] J. H. Kalir. Open web annotation as collaborative learning. *First Monday*, June 2019.
- [99] J. H. Kalir and J. Dean. Web annotation as conversation and interruption. *Media Practice and Education*, 19(1):18–29, 2018.
- [100] G. C. Kane. The evolutionary implications of social media for organizational knowledge management. *Information and Organization*, 27(1):37–46, 2017.
- [101] M. Kennedy. Open Annotation and Close Reading the Victorian Text: Using Hypothes.is with Students. *Journal of Victorian Culture*, 21(4):550–558, Dec 2016.
- [102] M. Klein, H. Van de Sompel, R. Sanderson, H. Shankar, L. Balakireva, K. Zhou, and R. Tobin. Scholarly context not found: One in five articles suffers from reference rot. *PLOS ONE*, 9(12):1–39, Dec 2014.
- [103] P. Klemperer. Network effects and switching costs: Two short essays for the new palgrave. *SSRN Electronic Journal*, 2006.
- [104] H. Kodama, F. Yano, S. Ninomija, Y. Sakai, and S. Ninomiya. A digital imaging processing method for gastric endoscope picture. In *2014 47th Hawaii International Conference on System Sciences*, volume 2, pages 277–282, Los Alamitos, CA, USA, Jan 1988. IEEE Computer Society.
- [105] L. Kreiling and C. Paunov. Knowledge co-creation in the 21st century. Organisation for Economic Co-Operation and Development (OECD), Jun 2021. <https://doi.org/10.1787/c067606f-en>; Accessed: October 12, 2021.
- [106] K. Król and D. Zdonek. Peculiarity of the bit rot and link rot phenomena. *Global Knowledge, Memory and Communication*, 69(1/2):20–37, Oct. 2019.
- [107] P. Kucherbaev, F. Daniel, S. Tranquillini, and M. Marchese. Crowdsourcing processes: A survey of approaches and opportunities. *IEEE Internet Computing*, 20(2):50–56, Mar 2016.
- [108] D. V. Kumar, B. T. S. Kumar, D. R. Parameshwarappa, and and. URLs link rot: Implications for electronic publishing. *World Digital Libraries - An International Journal*, 8(1), 2015.
- [109] D. M. J. Lazer, M. A. Baum, Y. Benkler, A. J. Berinsky, K. M. Greenhill, F. Menczer, M. J. Metzger, B. Nyhan, G. Pennycook, D. Rothschild, M. Schudson, S. A. Sloman, C. R. Sunstein, E. A. Thorson, D. J. Watts, and J. L. Zittrain. The science of fake news. *Science*, 359(6380):1094–1096, Mar 2018.
- [110] C. Lee, T. Yang, G. D. Inchoco, G. M. Jones, and A. Satyanarayan. Viral visualizations: How coronavirus skeptics use orthodox data practices to promote unorthodox science online. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, May 2021.
- [111] P. M. Leonardi. The social media revolution: Sharing and learning in the age of leaky knowledge. *Information and Organization*, 27(1):47 – 59, 2017.

- [112] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, Feb 1966. *Doklady Akademii Nauk SSSR*, V163 No4 845-848 1965.
- [113] P. Lévy. *Collective Intelligence: Mankind's Emerging World in Cyberspace*. Perseus Books, Cambridge, MA, USA, 1997.
- [114] P. Lévy. From social computing to reflexive collective intelligence: The IEML research program. *Information Sciences*, 180(1):71–94, Jan 2010.
- [115] D. Lewandowski. A three-year study on the freshness of web search engine databases. *Journal of Information Science*, 34(6):817–831, 2008.
- [116] S. J. Liebowitz and S. E. Margolis. Network externality: An uncommon tragedy. *Journal of Economic Perspectives*, 8(2):133–150, June 1994.
- [117] P. Lorenz-Spreen, S. Lewandowsky, C. R. Sunstein, and R. Hertwig. How behavioural sciences can promote truth, autonomy and democratic discourse online. *Nature Human Behaviour*, 4(11):1102–1109, June 2020.
- [118] V. Mallawaarachchi, L. Meegahapola, R. Madhushanka, E. Heshan, D. Meedeniya, and S. Jayarathna. Change detection and notification of web pages: A survey. *ACM Comput. Surv.*, 53(1), Feb. 2020.
- [119] T. W. Malone and M. S. Bernstein, editors. *Handbook of Collective Intelligence*. The MIT Press, Cambridge, MA, 2015.
- [120] K. Marissa. Using an online social annotation tool in a content-based instruction (cbi) classroom. *International Journal of TESOL Studies*, 3(2):5–23, 2021.
- [121] C. C. Marshall. Toward an ecology of hypertext annotation. In *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia : Links, Objects, Time and Space—structure in Hypermedia Systems: Links, Objects, Time and Space—structure in Hypermedia Systems*, HYPERTEXT '98, pages 40–49, New York, NY, USA, 1998. ACM.
- [122] H. Maurer. *Hyper-G Now Hyperwave: The Next Generation Web Solution*. Addison-Wesley, 1996.
- [123] G. Mulgan. *Big mind: How collective intelligence can change our world*. Princeton University Press, 2019.
- [124] H. Müller, T. Deselaers, T. M. Deserno, J. Kalpathy-Cramer, E. Kim, and W. Hersh. Overview of the imageclefmed 2007 medical retrieval and medical annotation tasks. In C. Peters, V. Jijkoun, T. Mandl, H. Müller, D. W. Oard, A. Peñas, V. Petras, and D. Santos, editors, *Advances in Multilingual and Multimodal Information Retrieval*, pages 472–491, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- [125] M. A. Musen. The protégé project: A look back and a look forward. *AI Matters*, 1(4):4–12, June 2015.
- [126] G. Navarro. A guided tour to approximate string matching. *ACM Comput. Surv.*, 33(1):31–88, mar 2001.

- [127] T. H. Nelson. Complex information processing: A file structure for the complex, the changing and the indeterminate. In *Proceedings of the 1965 20th National Conference*, ACM '65, page 84–100, New York, NY, USA, 1965. Association for Computing Machinery.
- [128] T. H. Nelson. *Literary Machines : The report on, and of, project xanadu concerning word processing, electronic publishing, Hypertext, thinkertoys, Tomorrow's intellectual revolution, and certain other topics including knowledge, education and freedom*. Swarthmore, Pa: Ted Nelson, 1981.
- [129] T. H. Nelson. Xanalogical structure, needed now more than ever: Parallel documents, deep links to content, deep versioning, and deep re-use. *ACM Comput. Surv.*, 31(4es):33–64, Dec 1999.
- [130] T. H. Nelson. Zigzag (tech briefing). In *Proceedings of the 12th ACM Conference on Hypertext and Hypermedia*, Hypertext '01, pages 261–262, New York, NY, USA, 2001. Association for Computing Machinery.
- [131] I. Nonaka, M. Kodama, A. Hirose, and F. Kohlbacher. Dynamic fractal organizations for promoting knowledge-based transformation – A new paradigm for organizational theory. *European Management Journal*, 32(1):137–146, 2014.
- [132] I. Nonaka and H. Takeuchi. *The Knowledge-creating Company: How Japanese Companies Create the Dynamics of Innovation*. Everyman's library. Oxford University Press, 1995.
- [133] I. Nonaka, R. Toyama, and T. Hirata. *Managing Flow: A Process Theory of the Knowledge-Based Firm*. Palgrave Macmillan UK, 2008.
- [134] A. Ntoulas, J. Cho, and C. Olston. What's new on the web? the evolution of the web from a search engine perspective. In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, page 1–12, New York, NY, USA, 2004. Association for Computing Machinery.
- [135] F. Oguz and W. Koehler. URL decay at year 20: A research note. *Journal of the Association for Information Science and Technology*, 67(2):477–479, 2016.
- [136] F. Oliveira and I. Ramos. Crowdsourcing: a tool for organizational knowledge creation. In *22nd European Conference on Information Systems*, 2014.
- [137] P. Oliveira and J. Rocha. Semantic annotation tools survey. In *2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 301–307, 2013.
- [138] T. O'Reilly. What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. MPRA Paper 4578, University Library of Munich, Germany, Mar. 2007.
- [139] S. Paavola, L. Lipponen, and K. Hakkarainen. Models of innovative knowledge communities and three metaphors of learning. *Review of Educational Research*, 74(4):557–576, 2004.
- [140] E. Pariser. *The Filter Bubble: What the Internet Is Hiding from You*. Penguin Group, UK, 2011.

- [141] V. Pattanaik. Tippanee - weave your own web. Chrome Web Store, 2018. <https://perma.cc/DDW4-AH85>; Accessed: December 2, 2021.
- [142] V. Pattanaik. Tippanee web annotator. GitHub, 2018. https://github.com/vpattanaik/Tippanee_Web_Annotator; Accessed: October 12, 2021.
- [143] V. Pattanaik. Web annotation test bench. GitHub, 2020. https://github.com/vpattanaik/WebAnnotation_TestBench; Accessed: October 12, 2021.
- [144] V. Pattanaik, A. Norta, M. Felderer, and D. Draheim. Systematic support for full knowledge management lifecycle by advanced semantic annotation across information system boundaries. In J. Mendling and H. Mouratidis, editors, *Information Systems in the Big Data Era*, pages 66–73, Cham, 2018. Springer International Publishing.
- [145] V. Pattanaik, I. Sharvadze, and D. Draheim. Framework for peer-to-peer data sharing over web browsers. In T. K. Dang, J. Küng, M. Takizawa, and S. H. Bui, editors, *Future Data and Security Engineering*, pages 207–225, Cham, 2019. Springer International Publishing.
- [146] V. Pattanaik, I. Sharvadze, and D. Draheim. A peer-to-peer data sharing framework for web browsers. *SN Computer Science*, 1(4), June 2020.
- [147] V. Pattanaik, M. Singh, P. Gupta, and S. Singh. Smart real-time traffic congestion estimation and clustering technique for urban vehicular roads. In *2016 IEEE Region 10 Conference (TENCON)*, pages 3420–3423, 2016.
- [148] V. Pattanaik, S. Suran, and D. Draheim. Enabling social information exchange via dynamically robust annotations. In *Proceedings of the 21st International Conference on Information Integration and Web-Based Applications & Services, iiWAS2019*, page 176–184, New York, NY, USA, 2019. Association for Computing Machinery.
- [149] V. Pattanaik, S. Suran, and S. Prabakaran. Inducing human-like motion in robots. In *Proceedings of the 6th IBM Collaborative Academia Research Exchange Conference (I-CARE) on I-CARE 2014*, I-CARE 2014, page 1–3, New York, NY, USA, 2014. Association for Computing Machinery.
- [150] T. Pearce, M. Maple, A. Shakeshaft, S. Wayland, and K. McKay. What is the co-creation of new knowledge? a content analysis and proposed definition for health interventions. *International Journal of Environmental Research and Public Health*, 17(7):2229, Mar. 2020.
- [151] S. A. Peious, S. Suran, V. Pattanaik, and D. Draheim. Enabling sensemaking and trust in communities: An organizational perspective. In *Proceedings of the 23rd International Conference on Information Integration and Web-Based Applications & Services, iiWAS '21*, page 1–9, New York, NY, USA, 2021. Association for Computing Machinery.
- [152] J. M. Perkel. Annotating the scholarly web. *Nature*, 528(7580):153–154, Dec. 2015.
- [153] J. M. Perkel. The trouble with reference rot. *Nature*, 521(7550):111–112, May 2015.

- [154] Pew Research Center. Many Experts Say Digital Disruption Will Hurt Democracy. Technical Report February, Pew Research Center: Internet, Science & Tech, 2020.
- [155] T. A. Phelps and R. Wilensky. Multivalent annotations. In C. Peters and C. Thanos, editors, *Research and Advanced Technology for Digital Libraries*, pages 287–303, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.
- [156] S. Pletinckx, K. Borgolte, and T. Fiebig. Out of sight, out of mind. In *Proceedings of the 28th ACM Conference on Computer and Communications Security*. ACM, Nov. 2021.
- [157] W. J. Potter. The state of media literacy. *Journal of Broadcasting & Electronic Media*, 54(4):675–696, 2010.
- [158] B. J. Regeer and J. F. Bunders. Knowledge co-creation: Interaction between science and society. *A Transdisciplinary Approach to Complex Societal Issues. Den Haag: Advisory Council for Research on Spatial Planning, Nature and the Environment/-Consultative Committee of Sector Councils in the Netherlands [RMNO/COS]*, 2009.
- [159] M. Rizun and V. G. Meister. Analysis of benefits for knowledge workers expected from knowledge-graph-based information systems. In S. Wrycza and J. Maślankowski, editors, *Information Systems: Research, Development, Applications, Education*, pages 25–39, Cham, 2017. Springer International Publishing.
- [160] C. A. Robert. Annotation for knowledge sharing in a collaborative environment. *Journal of Knowledge Management*, 13(1):111–119, Feb. 2009.
- [161] J. Salmons and L. Wilson, editors. *Handbook of Research on Electronic Collaboration and Organizational Synergy*. IGI Global, 2009.
- [162] R. Sanderson, P. Ciccicarese, and H. Van de Sompel. Designing the w3c open annotation data model. In *Proceedings of the 5th Annual ACM Web Science Conference, WebSci '13*, pages 366–375, New York, NY, USA, 2013. ACM.
- [163] R. Sanderson and H. Van de Sompel. Making web annotations persistent over time. In *Proceedings of the 10th Annual Joint Conference on Digital Libraries, JCDL '10*, page 1–10, New York, NY, USA, 2010. Association for Computing Machinery.
- [164] D. A. Scheufele and N. M. Krause. Science audiences, misinformation, and fake news. *Proceedings of the National Academy of Sciences*, 116(16):7662–7669, 2019.
- [165] C. O. Schneble, B. S. Elger, and D. Shaw. The cambridge analytica affair and internet-mediated research. *EMBO reports*, 19(8):e46579, 2018.
- [166] D. Schneckenberg. Web 2.0 and the empowerment of the knowledge worker. *Journal of Knowledge Management*, 13(6):509–520, Oct. 2009.
- [167] S. M. Schneider and K. A. Foot. The web as an object of study. *New Media & Society*, 6(1):114–122, 2004.
- [168] R. Service. DNA could store all of the world's data in one room. *Science*, Mar. 2017.
- [169] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu. Fake News Detection on Social Media: A Data Mining Perspective. *SIGKDD Explor. Newsl.*, 19(1):22–36, Sep 2017.

- [170] O. Solon. Tim Berners-Lee on the future of the web: 'The system is failing'. The Guardian, Nov 2017.
<https://perma.cc/8XKH-U8S4>; Accessed: October 12, 2021.
- [171] L. Stein. Genome annotation: from sequence to biology. *Nature Reviews Genetics*, 2(7):493–503, July 2001.
- [172] U. Straccia, N. Lopes, G. Lukacsy, and A. Polleres. A general framework for representing and reasoning with annotated semantic web data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 24(1):1437–1442, Jul. 2010.
- [173] Y. Sun and F. Gao. Web annotation and threaded forum: How did learners use the two environments in an online discussion? *Journal of Information Technology Education: Innovations in Practice*, 13:069–088, 2014.
- [174] S. Suran, V. Pattanaik, and D. Draheim. Communitycare: Tackling mental health issues with the help of community. In *Proceedings of the 22nd International Conference on Information Integration and Web-Based Applications & Services, iiWAS '20*, page 377–382, New York, NY, USA, 2020. Association for Computing Machinery.
- [175] S. Suran, V. Pattanaik, and D. Draheim. Frameworks for collective intelligence: A systematic literature review. *ACM Comput. Surv.*, 53(1), Feb. 2020.
- [176] S. Suran, V. Pattanaik, and D. Malathi. Discovering shortest path between points in cerebrovascular system. In *Proceedings of the 6th IBM Collaborative Academia Research Exchange Conference (I-CARE) on I-CARE 2014*, I-CARE 2014, page 1–3, New York, NY, USA, 2014. Association for Computing Machinery.
- [177] S. Suran, V. Pattanaik, M. Singh, P. Gupta, and P. Gupta. Brain imaging procedures and surgery techniques: Past, present and future. *International Journal of Bio-Science and Bio-Technology*, 9(3):23–34, June 2017.
- [178] S. Suran, V. Pattanaik, S. B. Yahia, and D. Draheim. Exploratory analysis of collective intelligence projects developed within the eu-horizon 2020 framework. In N. T. Nguyen, R. Chbeir, E. Exposito, P. Anierté, and B. Trawiński, editors, *Computational Collective Intelligence*, pages 285–296, Cham, 2019. Springer International Publishing.
- [179] C. Tikkinen-Piri, A. Rohunen, and J. Markkula. Eu general data protection regulation: Changes and implications for personal data collecting companies. *Computer Law & Security Review*, 34(1):134–153, 2018.
- [180] A. N. Tump, T. J. Pleskac, and R. H. J. M. Kurvers. Wise or mad crowds? the cognitive mechanisms underlying information cascades. *Science Advances*, 6(29):eabb0266, 2020.
- [181] H. Tyagi, S. Suran, and V. Pattanaik. Weather - temperature pattern prediction and anomaly identification using artificial neural network. *International Journal of Computer Applications*, 140(3):15–21, Apr. 2016.
- [182] P. Voigt and A. von dem Bussche. *The EU General Data Protection Regulation (GDPR)*. Springer International Publishing, 2017.

- [183] G. von Krogh, I. Nonaka, and L. Rechsteiner. Leadership in organizational knowledge creation: A review and framework. *Journal of Management Studies*, 49(1):240–277, 2012.
- [184] D. Wagner, G. Vollmar, and H.-T. Wagner. The impact of information technology on knowledge creation. *Journal of Enterprise Information Management*, 27(1):31–44, Feb. 2014.
- [185] J. West, A. Salter, W. Vanhaverbeke, and H. Chesbrough. Open innovation: The next decade. *Research Policy*, 43(5):805 – 811, 2014. Open Innovation: New Insights and Evidence.
- [186] S. Wise, R. A. Paton, and T. Gegenhuber. Value co-creation through collective intelligence in the public sector. *VINE Journal of Information and Knowledge Management Systems*, 42(2):251–276, May 2012.
- [187] K. Wodzicki, E. Schwämmlein, and J. Moskaliuk. “actually, i wanted to learn”: Study-related knowledge exchange on social networking sites. *The Internet and Higher Education*, 15(1):9–14, Jan. 2012.
- [188] K.-P. Yee. Critlink: Advanced hyperlinks enable public annotation on the web. *University of California, Berkeley*, 2002.
- [189] A. X. Zhang and J. Cranshaw. Making Sense of Group Chat through Collaborative Tagging and Summarization. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–27, Nov 2018.
- [190] W. Zhang, B. Paudel, L. Wang, J. Chen, H. Zhu, W. Zhang, A. Bernstein, and H. Chen. Iteratively learning embeddings and rules for knowledge graph reasoning. In *The World Wide Web Conference, WWW '19*, page 2366–2377, New York, NY, USA, 2019. Association for Computing Machinery.
- [191] K. Zhou, C. Grover, M. Klein, and R. Tobin. No more 404s: Predicting referenced link rot in scholarly articles for pro-active archiving. In *Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL '15*, page 233–236, New York, NY, USA, 2015. Association for Computing Machinery.
- [192] X. Zhou and R. Zafarani. A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Comput. Surv.*, 53(5), Sept. 2020.
- [193] X. Zhu, B. Chen, R. M. Avadhanam, H. Shui, and R. Z. Zhang. Reading and connecting: using social annotation in online classes. *Information and Learning Sciences*, 121(5/6):261–271, June 2020.
- [194] A. Zimmermann, N. Lopes, A. Polleres, and U. Straccia. A general framework for representing, reasoning and querying with annotated semantic web data. *Journal of Web Semantics*, 11:72–95, 2012.
- [195] J. Zittrain, K. Albert, and L. Lessig. Perma: Scoping and addressing the problem of link and reference rot in legal citations. *Legal Information Management*, 14:88–99, 2014.

Acknowledgements

First and foremost, I am extremely grateful to my supervisor Dirk Draheim for his invaluable advice, continuous support, and patience during my PhD study. His immense knowledge and plentiful experience have encouraged me through the duration of my academic research. I am grateful to my alma mater Tallinn University of Technology, for having made this work possible.

Deep from my heart, I would like to express my gratitude to my parents, my wife, and my friends. Without their tremendous understanding and encouragement in the past few years, it would not have been possible for me to complete my study.

Abstract

Robust Web Annotations in Support of Knowledge Co-Creation

Web technologies have enabled citizens of the world to come together, interact and collaborate in unprecedented ways. Advancements in the field has empowered web users to transition from being content consumers to content creators, and has allowed them to interact and work together with other web users from around the world. This has led to the development of numerous novel applications ranging from domains such as culture and democracy, to science and education. One such class of applications are web annotation systems. These systems have enabled web users to highlight, store, critique (i.e., through comments) and share pieces of information over web browsers; thereby enabling communities (particularly, from academia and research) to exchange and create knowledge by conversing over the entire web. This enthusiasm around the application of web annotation systems in knowledge sharing and learning, is now being challenged by the web's ephemerality. Furthermore, as argued by researchers, lack of system features supporting new knowledge creation and learning models, is preventing the adoption of web annotation systems as a primary tool for learning. That said, the work presented in this dissertation seeks to address both of these issues, through four distinct contributions.

The work presented in Chapter 3 investigates several well-adapted frameworks and models for knowledge transformation and creation, and introduces a novel framework for web annotation activities in support of knowledge co-creation. The framework consists of multiple textual and semantic web annotation activities, segregated into four processes, namely socialization, externalization, combination and internalization (based on Nonaka's SECI model). The proposed activities are further segregated into two levels, and are explained both from the point-of-view of a (non-expert) web user and a knowledge expert. The proposed annotation activities are designed into a novel web annotation system in Chapter 5.

The research in Chapter 4 is aimed at the development of an anchoring algorithm that is more robust against ephemeral web content. As evident from literature, current state-of-the-art in anchoring algorithms are unable to keep up the ephemeral and transient nature of content on the web. This has led to concerns about annotations being orphaned over time, which can be critical, especially when annotations are being used to create and share knowledge. Considering this issue, in Chapter 4, a novel anchoring algorithm is proposed. The algorithm utilizes three sets of information, namely the annotated text, its surrounding context (i.e., in textual form) and its structure, and generates anchors that are more resilient to both textual and structural decays. The robustness of the proposed algorithm was evaluated by comparing it against Hypothes.is' fuzzy anchoring algorithm. During the initial experimentation, the proposed algorithm was found to be more robust than the state-of-the-art. Considering the limitation of the small number of annotations used during the experimentation, a more rigorous alternative approach (i.e., test bench) to evaluate the robustness of anchoring algorithms was proposed in Chapter 6.

The relevance of the work presented in Chapter 5 is twofold. This chapter introduces a novel web annotation system called Tippanee. The platform is built around the novel anchoring algorithm presented in Chapter 4, and is designed with system features based on the annotation activities proposed in Chapter 3. Developed as client-side browser extension, the annotation tool enables its users to not only create and share textual annotations, but also allows them to archive, link, and visualize said annotations on their local machines. Through its novel feature sets, the annotation system aims to empower its users by supporting knowledge co-creation activities. The system's features were evaluated by conducting a lab experiment with participants, and most participants reported

that they found the system both easy-to-use and useful. As an outcome of the lab experiment, additional insights into the behaviour of web annotation system users were identified.

Finally, Chapter 6 addresses the challenge of empirically and reproducibly evaluating the robustness of anchoring algorithms, and tests the hypothesis that considering annotation structure when generating anchors can make annotations more robust against ephemeral web content. A major contribution of the research presented in the chapter is the (first of its kind) web annotation test bench that simulates various forms of textual and structural web page decays, and thus allows its users to compare the robustness and accuracy of web annotation anchoring algorithms. Meticulously designed by selecting several real-world web pages, and manually generated annotations, the test bench provides its users with 1,110 annotations and their 11,100 variations. By viewing the various decayed annotations through a web annotation system, users are able to evaluate the robustness and accuracy of the system being tested. By comparing the novel anchoring algorithm presented in Chapter 4 against the state-of-the-art (i.e., Hypothes.is), when using the test bench, the novel algorithm proposed in this /thesis was found to be 9.5% more robust and 12.5% more accurate.

Combined the four contributions presented in this dissertation add to the body of knowledge on web annotation systems, and provide insights for development of robust and stable web annotation systems that can support knowledge creation and exchange in today's dynamic, collaborative environments. Each of the research contributions by themselves provide a first step towards future improvements and investigations into the use of web annotations for knowledge management.

Kokkuvõte

Töökindlad veebiannotatsioonid teadmiste ühisloome toetamiseks

Veebitehnoloogiad võimaldavad inimestel ühenduda, suhelda ja koostööd teha. Valdkonnana edusammud on andnud veebikasutajatele võimaluse muutuda sisu tarbijast sisu loojaks ning suhelda ja töötada koos teiste veebikasutajatega üle kogu maailma. See on viinud arvukate uudsete lahenduste väljatöötamiseni kultuuri-, demokraatia-, teadus- ja haridusvaldkondades. Veebiannotatsioonid on üks taoline lahendus. Veebiannotatsioonid võimaldavad veebikasutajatel esile tõsta, salvestada, kritiseerida (nt kommenteerides) ja jagada teavet veebi kaudu. See võimaldab kogukondadel, eelkõige akadeemilistes ja teaduslikes ringkondades, veebis vahetada ja luua teadmisi. Veebi muutlikkus seab aga kahtluse alla veebiannotatsioonide-süsteemide rakendamise teadmiste jagamisel ja õppeprotsessis. Lisaks väidavad teadlased, et teadmiste loomist ja õppeprotsessi toetavate funktsioonide puudumine takistab veebiannotatsioonide süsteemi kasutuselevõttu õppimise peamise vahendina. Doktoritöös käsitletakse mõlemat kõnealust küsimust nelja erineva teema kaudu.

Kolmandas peatükis uuritakse mitmeid hästitoimivaid raamistikke ja mudeleid teadmiste ümberkujundamiseks ja loomiseks ning tutvustatakse uudset, ühisloomet toetavat raamistikku veebiannotatsioonide lisamiseks. Raamistik koosneb mitmest tekstilise ja semantilise veebi märkimistegevusest, mis on jaotatud neljaks protsessiks: sotsialiseerimine, eksternaliseerimine, kombineerimine ja internaliseerimine (põhineb Nonaka SECI-mudelil). Väljapakutud tegevused on omakorda jaotatud kaheks tasandiks ning neid selgitatakse nii mitte-ekspertidest veebikasutajate kui ka ekspertide seisukohalt. Pakutud märkimistegevustest ehitatakse viiendas peatükis uus veebi märkimissüsteem.

Neljandas peatükis esitatud uuringu eesmärk on töötada välja ankurdamisalgoritm, mis on senisest vastupidavam muutliku veebisisu suhtes. Kirjandusest selgub, et praegused ankurdamisalgoritmide ei suuda pidada sammu veebisisu muutlikusega. Seepärast võivad märkused aja jooksul kasutusest kaduda, mis on aga probleem, kui märkmeid taetakse kasutada teadmiste loomiseks ja jagamiseks. Sellest lähtuvalt pakutakse neljandas peatükis välja uus ankurdamisalgoritm. Algoritmi töö põhineb kolmel infokogumil: kommenteeritud tekstil, seda ümbritseval kontekstil (st tekstilisel kujul) ja selle struktuuril. Algoritm genereerib ankurdusi, mis on paindlikumad nii tekstilise kui ka struktuurilise hääbumise suhtes. Välja töötatud algoritmi töökindlust hinnati, võrreldes seda Hypothes.is'i programmi ankurdamisalgoritmiga. Esialgsete katsete käigus leiti, et välja töötatud uus algoritm on töökindlam kui praegune algoritm. Kuna esialgses katses oli analüüsimiseks liiga vähe märkuseid, pakuti kuuendas peatükis välja rangem alternatiivne lähenemisviis (katseplatvorm) ankurdamisalgoritmide töökindluse hindamiseks.

Viiendas peatükis tutvustatakse uudset veebi märkustesüsteemi Tippanee. Platvormi aluseks on neljandas peatükis esitatud uus ankurdamisalgoritm, mis on loodud kolmandas peatükis esitatud märkimistegevuste põhjal. Klientidepoolse veebilehitseja laiendusena välja töötatud märkuste lisamise vahend võimaldab kasutajatel mitte ainult luua ja jagada tekstilisi märkusi, vaid neid ka oma arvutisse salvestada, linkida ja visualiseerida oma arvutis. Uudsete funktsioonide abil on märkimissüsteemi eesmärk anda oma kasutajale rohkem võimalusi ja toetada teadmiste ühisloomet. Süsteemi funktsioone hinnati laborikatsete abil ning enamik osalejaid pidas süsteemi nii hõlpsasti kasutatavaks kui ka kasulikuks. Laborikatsetuse tulemusena saadi täiendavaid teadmisi veebi märkimissüsteemi kasutajate käitumise kohta.

Kuuendas peatükis käsitletakse ankurdamisalgoritmide töökindluse empiirilise ja rep-

rodutseeritavuse hindamise väljakutset ning testitakse hüpoteesi, et ankurdamisstruktuuri arvestamine ankurdamise genereerimisel võib muuta ankurdamise töökindlamaks muutliku veebisisu suhtes. Peatükis esitatud uurimuse peamine panus on (esimene omataoline) veebiannotatsioonide katsekeskkond, mis simuleerib erinevaid tekstilise ja struktuurilise veebilehe lagunemise vorme ja võimaldab seega selle kasutajatel võrrelda veebiannotatsioonide ankurdamisalgoritmide töökindlust ja täpsust. Katsekeskkond on hoolikalt koostatud mitme reaalse veebilehe ja käsitsi loodud märkuste valimise teel, ning see pakub kasutajale 1110 märkust ja nende 11 100 varianti. Erinevate märkmete vaatamine süsteemi kaudu võimaldab kasutajatel hinnata testitava süsteemi töökindlust ja täpsust. Neljandas peatükis esitatud uut ankurdamisalgoritmi võrreldi tiptasemel Hypothes.is'iga ning leiti, et käesolevas töös välja pakutud uus algoritm on 9,5 % töökindlam ja 12,5 % täpsem.

Doktoritöös esitatud neli teemat täiendavad veebi märkimissüsteemide alaseid teadmisi ja annavad ülevaate töökindlate ja stabiilsete veebiannotatsioonide-süsteemide arendamiseks, toetamaks teadmiste loomist ja vahetamist tänapäeva dünaamilistes koostöökeskkondades. Iga teadustöös esitatud teema on esimene samm tulevaste täiustuste ja uuringute jaoks, mis käsitlevad veebiannotatsioonide kasutamist teadmiste haldamiseks.

Appendix 1

I

V. Pattanaik, A. Norta, M. Felderer, and D. Draheim. Systematic support for full knowledge management lifecycle by advanced semantic annotation across information system boundaries. In J. Mendling and H. Mouratidis, editors, *Information Systems in the Big Data Era*, pages 66–73, Cham, 2018. Springer International Publishing



Systematic Support for Full Knowledge Management Lifecycle by Advanced Semantic Annotation Across Information System Boundaries

Vishwajeet Pattanaik¹ , Alex Norta¹, Michael Felderer²,
and Dirk Draheim¹ 

¹ Tallinn University of Technology, Tallinn, Estonia
vpattanaik@gmail.com, alex.norta.phd@ieee.org, draheim@acm.org

² University of Innsbruck, Innsbruck, Austria
michael.felderer@uibk.ac.at

Abstract. In today's organizations, there exist a variety of information-system paradigms to support the diverse organizational needs that reside on different levels of operations, tactics and strategies on the one hand, and are subject to different work styles on the other hand. This makes it challenging to design knowledge management systems that can be integrated into heterogeneous information-system applications. In this paper, we present Tippianee that allows users to create and manage semantic annotations over dynamically generated contents across the boundaries of arbitrary information-system applications. We further show, how these advanced semantic annotation capabilities can be systematically exploited in the full knowledge-management lifecycle. We explain, how the essentially important transformation of tacit-explicit-tacit knowledge corresponds to the reorganization of semantic schemata by a mature editorial process.

Keywords: Enterprise resource planning · Knowledge management
Metadata management solutions · Semantic annotations
Social software

1 Introduction

Knowledge management in organizations has been a critical subject of research over decades. Organizations must continuously come up with innovative products and solutions, while seeking and creating new knowledge, and embracing wisdom from society [1–4]. Innovations are becoming more socially dynamic, and rely on simultaneous creation and exploitation of knowledge [5].

While, the massive growth of social media platforms has made it easier to retrieve knowledge from society using crowdsourcing practices [6,7] such as crowd-voting, crowd-funding and microwork. However, designing knowledge

management systems and integrating them into increasing number of information systems applications is becoming a challenging task. On one hand, enterprises are embracing Open Innovation practices [8], while on the other hand knowledge sharing among employees within these enterprises isn't necessarily flourishing [9].

This is because organizations rely on a variety of Information System (IS) applications to support their business activities. While some organizations design their systems and tools themselves, others acquire them from third-party experts. The latter are often independent legacy systems and packages which generally result in heterogeneous software ecosystems. In most cases, large enterprises that build their information systems in-house are able to develop, integrate and maintain their Knowledge Management Systems (KMSs), whereas the same is often impossible for small to medium enterprises (SMEs) as they lack the resources required to perform core KMS activities [10]. Moreover, KMSs are generally designed for a predetermined workflow and lack features to support informal person-to-person communication. Due to which, employees often resort to asking colleagues or improvising in the absence of guidance, habitually missing best practices and repeating mistakes [11]. Employees also might not be prepared to share information in order to protect their jobs or, they might be too busy and may choose not to funnel information into such systems [12].

Considering the aforementioned issues, we derive the following challenges: (1) How to design an independent-interoperable knowledge management tool for organizations with heterogeneous software ecosystems? (2) How to enable knowledge management in a social setting, while using organizational information systems? (3) Can describing things on-the-fly motivate employees to share their knowledge? If so, how? (4) What kind of moderation process would be required to consolidate the knowledge shared by employees?

The Tiptanee platform intends to resolve the above mentioned challenges by promoting systematic knowledge creation and management independent of the underlying information systems. The platform is founded on the notion that organizations are social entities and in order to become knowledge creating companies, organizations must encourage knowledge creating and transforming activities while allowing social interactions among employees [1, 11, 13, 14]. Tiptanee^{1,2} is designed as a lightweight, user-friendly Google Chrome browser extension that allows users to highlight, add, link and share annotations hooked onto elements and processes within HTML documents. Its novel and robust anchoring algorithm detects and reattaches anchors on arbitrarily generated web pages. The platform's semantic annotation features are inspired by the SECI model (socialization, externalization, combination, internalization) of knowledge dimensions [1], therefore it encourages knowledge creation, conversion and transfer. The system enables users to share their tacit knowledge by converting it into semantic descriptions attached to annotated content. Furthermore, users are allowed to create their own semantic vocabularies, making it easier for them to

¹ <http://tinyurl.com/tiptanee>.

² <https://github.com/vpattanaik/Tiptanee>.

express their knowledge. Lastly, the system utilizes a moderation process that is somewhat social and includes not just knowledge experts but also the user pool.

The remainder of this paper is structured as follows. Section 2 gives an overview of the related work. Section 3 details the Tippanee platform and its feature from different user perspectives. Finally, Sect. 4 presents a critical discussion about the current stage of our research and concludes the paper.

2 Related Work

2.1 Social Weaver

The core intention of the Social Weaver platform is to enrich existing enterprise applications by weaving them together with end-user applications. The platform enables end users to weave snippets of social software features such as bookmarks, comments and wikis onto the workflow of enterprise applications [14, 15]. Unfortunately, since Social Weaver’s anchoring algorithm is based on string search, it cannot handle changes in web content. Inspired by Social Weaver, the Tippanee platform extends these social features to enterprise applications while being able to handle content changes, thanks to its novel anchoring approach.

2.2 Fuzzy Anchoring

Used by the annotation platforms Hypothesis³ and Genius⁴, the Fuzzy Anchoring approach is a combination of fuzzy string matching and XML Path (XPath) matching. The approach provides a novel solution for attaching annotations on HTML documents but focuses more on string matching than on structure matching. The fuzzy string matching logic used in the algorithm is a combination of search and compare⁵; and although the algorithm works efficiently on text contents with minor changes, however dramatic changes in content or webpage structure renders the approach useless.

2.3 RDF and Schema.org

Resource Description Framework (RDF) is a data model for representing information about things on the Web. Originally designed as a data model for metadata, the framework allows developers to add metadata about web content. It was designed as a tool for knowledge representation, to assist machine learning algorithm such as search engines and help them better understand, link and present web content⁶. Derived from RDF Schema, Schema.org creates, maintains and promotes a common set of schemas for structured data markup. The

³ <https://web.hypothes.is/>.

⁴ <https://genius.com/web-annotator>.

⁵ <https://web.hypothes.is/blog/fuzzy-anchoring/>.

⁶ <https://www.w3.org/TR/rdf11-primer/>.

shared vocabulary provided by Schema.org are developed by an open community process. This shared vocabulary makes it easier for webmasters and developers to add descriptions to web resources⁷.

3 Tippane

Tippane with its semantic annotation capabilities and design features is meant to encourage employees to create, transform and share knowledge within the organization; following the principles of Knowledge Spiral proposed by Nonaka [1]. The platform supports knowledge creation and facilitates social interactions by means of a user-friendly interface, combined with a robust anchoring algorithm and semantic capabilities. We describe of these features in detail in the following sections.

3.1 User Interface

The Tippane platform is realized as a Google Chrome browser extension which is designed to be independent i.e., the extension can create and reattach annotations without communicating with the server. Similar to Hypothesis⁸, the Tippane platform functions on a superficial layer above the web page allowing it to be interoperable. When the user opens a webpage on the browser, the extension injects Tippane's dashboard and its supporting scripts into the HTML Document. The user interface allows users to select web contents even at the sentence level. Annotation are displayed on the right side of the screen, within Tippane's dashboard. Users can add multiple notes to the annotated text, they can link annotations from different webpages, reconstruct orphaned anchors and add semantic descriptions to annotations. Figure 1, illustrates a snapshot of Tippane's dashboard.

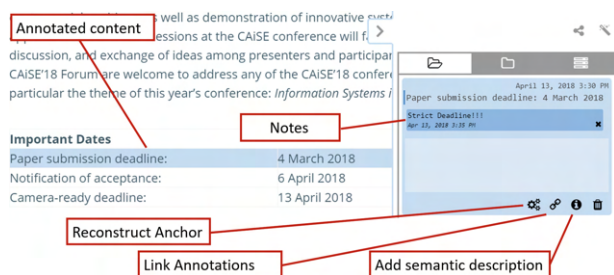


Fig. 1. A section of Tippane's dashboard, with added note and the 'Describe Anchor', 'Link Notes' and 'Reconstruct Anchor' buttons on the bottom right

⁷ <http://schema.org/docs/datamodel.html>.

⁸ <https://web.hypothes.is/about/>.

3.2 Anchoring Algorithm

To generate stable anchors, Tiptanee’s anchoring algorithm analyses HTML Document Object Model (DOM) elements using a predefined set of attributes. When reattaching annotations, the algorithm uses an edit distance approach [16] based on attribute matches and mismatches. This tree matching approach allows for robust anchoring of annotations even on arbitrarily generated webpages. Furthermore, stored anchor information enables preservation and reconstruction of annotations even when they are orphaned.

3.3 Semantics

Tiptanee’s semantic description feature allows users to describe annotations using Schema.org’s metadata data model. The feature is designed keeping two different categories of users in mind. The first category of user is an *employee*. An employee can add simple textual notes and semantic descriptions to annotated content using predefined semantic types and properties. An employee is also allowed to create new semantic types/properties and then describe annotations using the same. The second category of user is the *organization’s knowledge expert* hereby referred as the *moderator*. The moderator primarily maintains the knowledge created within the organization. Moderators can use Tiptanee as a general employee, however they can also review old-new semantic types and properties. They are allowed to standardize new semantic types and properties based on the ones created by employees or, on the basis of the organizations requirements. Furthermore, moderators are allowed to add, collaborate and restructure the semantic types/properties based on other employees suggestions or the organizations demands.

Features for Employees. The Tiptanee platform facilitates employees with two modes: *Plain Annotation Mode* and *Descriptive Mode*. The *Plain Annotation Mode* allows employees to add simple textual notes to annotations. This allows users to highlight and share sections of interests within the Information System. The mode can also be utilized by managers and supervisors to share important announcements and add descriptions to process that might be less understood by users. The feature supports Tiptanee’s motivation of allowing social interactions within Information Systems, without the need of switching between applications or services. For instance, if a manager would like to inform all his employees of a new feature or a change in the Information System, the manager would usually send an email to the employees informing them about the same. However when using Tiptanee, the manager could simply add an annotation to a content on the Information System’s homepage and make the annotation viewable to all employees who access the system. This could allows employees to have discussions or ask questions over the same annotation. Since employees can reply to annotations on the fly, the annotations would promote social interactivity. Also, new employees who join the organization at a later period could simply read the annotations and the replies and could update themselves without actually asking for help.

The *Descriptive Mode* is an extension to the *Plain Annotation Mode* and allows users to describe their annotations using semantic descriptions. The Tippanee platform utilizes Schema.org' metadata data model and therefore describes semantic descriptions as 'types' and 'properties'. 'Types' generally represent an item/thing and can have several properties that describe them. The 'properties' are keys to values but can also be described using a 'type'. This means that values of a property could also be complex values which are bundles of other 'properties'.

For instance, lets consider the following description of an organization:
organization: {companyname: *text*, sameas: *text*, ownedby: {person: *firstname: text, lastname: text*}}.

The description starts with the type 'organization', which has 'companyname', 'sameas', 'ownedby' as its properties. The properties 'companyname' and 'sameas' are described as simple key-value pairs where the values are of text type. However, the 'ownedby' property is further described using the type 'person', which is then described through its properties 'firstname' and 'lastname'. These are further described using key-value pairs with text type values.

The *Descriptive Mode* is further distinguished into two modes: *Strict Descriptive Mode* and *Flexible Mode*. In the *Strict Descriptive Mode*, the user can only use already available types and properties, whereas the *Flexible Mode* allows customizations to previously available semantic vocabulary, further enhancing the user's knowledge creation capabilities. While creating new vocabularies the user can either decide to do it systematically or as ad-hoc. Adding *Systematic* vocabulary means that the user has to clearly define the added types and properties. The user can either add properties to predefined types and then add values to the new properties; or introduce new types, define their properties and then add values. This way of adding vocabulary is meant for users who understand the organizations standard semantic vocabulary but can't find the appropriate types/properties required to describe specific processes. Adding *ad-hoc* vocabulary means that the user can either simply add key-value pairs independent of the available types and properties, or can add just a key or just a value. This way of adding vocabulary is meant for users who just want to describe their annotations semantically but are not interested in providing or at the time can't think of an appropriate description. Adding ad-hoc vocabulary might also be useful in situations where a user might want to make a quick suggestion, but does not want to waste time looking through the available vocabulary.

Features for Knowledge Experts. The knowledge expert or moderator is a person who understands the semantic vocabulary of the organization and is responsible for updating and maintaining the vocabulary available within the Tippanee platform. It is the expert's responsibility to study the new vocabularies suggested by other employees and organize them according to the organizations requirements. These semantic vocabularies may change from time-to-time depending upon the organizations goals, agendas and processes. However, since the consolidation process suggested by Tippanee is based on social processes, the

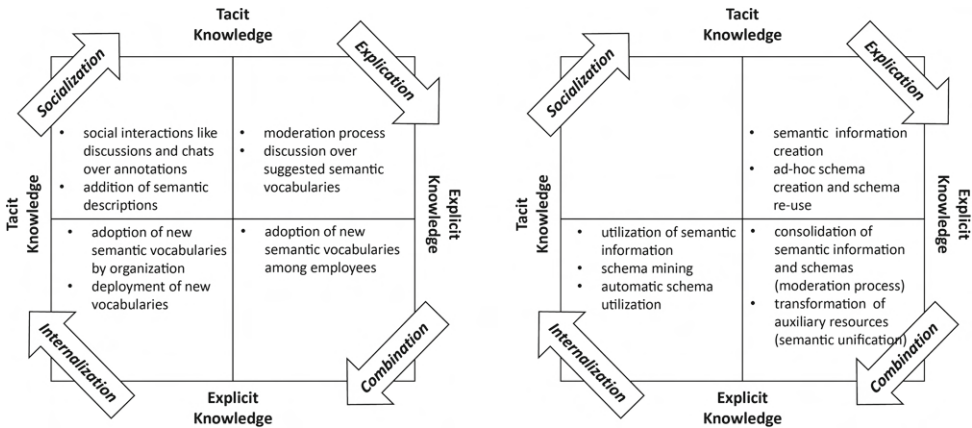


Fig. 2. Lower level (*left*) and higher level (*right*) annotation activities in Tippanee in coordination with the SECI model.

platform tries to include the community in improving the semantic vocabulary; the moderator can look at suggestions provided employees in order to improve the already available semantic vocabulary.

It is vital and interesting to understand that Tippanee’s semantic description feature utilizes Nonaka’s SECI model at two different levels. At the lower level, the *Plain Annotation Mode* and *Strict Descriptive Mode* allow users to share knowledge in a more structured way, whereas at the higher level the *Flexible Mode* allows sharing of unstructured/semi-structured knowledge; as illustrated in Fig. 2.

4 Conclusion

The Tippanee platform is designed with the intention to introduce Nonaka’s concept of a Knowledge Creating Company into organizations’ heterogeneous software ecosystems. Currently, the platform is in its early stage of development. The current version of Tippanee available on Chrome Web Store, illustrates our novel-robust-interoperable anchoring approach. The extension allows individual users to add, link, visualize and semantically describe annotations. However, the social features and moderation process are still in development.

On completion, Tippanee would allow users to create, share and transform their tacit knowledge into explicit knowledge by the means of semantics. The platform’s social environment would encourage users to participate in knowledge sharing and would allow them to semantically annotate and share process descriptions, further encouraging them to create and share knowledge, not just with work-groups but within the organizations’ as whole. Tippanee would further enhance the social aspects of knowledge creation through its moderation process. It would be interesting to see how employees from different departments would

participate as a larger community. Also this would encourage interdepartmental knowledge sharing, enhancing both the employees' and the organizations' knowledge.

References

1. Nonaka, I., Takeuchi, H.: *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press, Oxford (1995). Everyman's library
2. Nonaka, I., Toyama, R., Hirata, T.: *Managing Flow: A Process Theory of the Knowledge-Based Firm*. Palgrave Macmillan, Basingstoke (2008)
3. von Krogh, G., Nonaka, I., Rechsteiner, L.: Leadership in organizational knowledge creation: a review and framework. *J. Manag. Stud.* **49**(1), 240–277 (2012)
4. Leonardi, P.M.: The social media revolution: sharing and learning in the age of leaky knowledge. *Inf. Organ.* **27**(1), 47–59 (2017)
5. Nonaka, I., Kodama, M., Hirose, A., Kohlbacher, F.: Dynamic fractal organizations for promoting knowledge-based transformation a new paradigm for organizational theory. *Eur. Manag. J.* **32**(1), 137–146 (2014)
6. Oliveira, F., Ramos, I.: Crowdsourcing: a tool for organizational knowledge creation. In: *Twenty Second European Conference on Information Systems* (2014)
7. Kucherbaev, P., Daniel, F., Tranquillini, S., Marchese, M.: Crowdsourcing processes: a survey of approaches and opportunities. *IEEE Internet Comput.* **20**(2), 50–56 (2016)
8. West, J., Salter, A., Vanhaverbeke, W., Chesbrough, H.: Open innovation: the next decade. *Res. Policy* **43**(5), 805–811 (2014). Open innovation: new insights and evidence
9. Wang, S., Noe, R.A., Wang, Z.M.: Motivating knowledge sharing in knowledge management systems: a quasifield experiment. *J. Manag.* **40**(4), 978–1009 (2014)
10. Nunes, M.B., Annansingh, F., Eaglestone, B., Wakefield, R.: Knowledge management issues in knowledge-intensive smes. *J. Doc.* **62**(1), 101–119 (2006)
11. Hemsley, J., Mason, R.M.: Knowledge and knowledge management in the social media age. *J. Organ. Comput. Electron. Commer.* **23**(1–2), 138–167 (2013)
12. Kaplan, A.M., Haenlein, M.: Users of the world, unite! The challenges and opportunities of social media. *Bus. Horiz.* **53**(1), 59–68 (2010)
13. Wagner, D., Vollmar, G., Wagner, H.T.: The impact of information technology on knowledge creation: an affordance approach to social media. *J. Enterp. Inf. Manag.* **27**(1), 31–44 (2014)
14. Draheim, D., Felderer, M., Pekar, V.: Weaving social software features into enterprise resource planning systems. In: Piazzolo, F., Felderer, M. (eds.) *Novel Methods and Technologies for Enterprise Information Systems*. LNISO, vol. 8, pp. 223–237. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07055-1_18
15. Draheim, D.: The present and future of large-scale systems modeling and engineering. In: Dang, T.K., Wagner, R., Küng, J., Thoai, N., Takizawa, M., Neuhold, E. (eds.) *FDSE 2016*. LNCS, vol. 10018, pp. 355–370. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48057-2_25
16. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals. *Sov. Phys. Dokl.* **10**, 707 (1966)

Appendix 2

II

V. Pattanaik, S. Suran, and D. Draheim. Enabling social information exchange via dynamically robust annotations. In *Proceedings of the 21st International Conference on Information Integration and Web-Based Applications & Services, iiWAS2019*, page 176–184, New York, NY, USA, 2019. Association for Computing Machinery

Enabling Social Information Exchange via Dynamically Robust Annotations

Vishwajeet Pattanaik
Information Systems Group,
Tallinn University of Technology
Tallinn, Estonia
vishwajeet.pattanaik@taltech.ee

Shweta Suran
Information Systems Group,
Tallinn University of Technology
Tallinn, Estonia
shweta@taltech.ee

Dirk Draheim
Information Systems Group,
Tallinn University of Technology
Tallinn, Estonia
dirk.draheim@taltech.ee

ABSTRACT

With the emergence of new web paradigms, we currently see a tremendous increase in interest in different applications of the social web. However, this rising interest in social platforms has also led to the rise of numerous new challenges, especially issues like fake-news, filter-bubble, and web-page-decay. Motivated by these issues, we propose a novel DOM-oriented edit distance anchoring approach that enables stable tracking of ephemeral web content. We argue that such a stable anchoring approach could indeed foster the creation of a browser-based crowdsourcing information system that could help us tackle rising issues on the web. Building on this hypothesis, we present a new web annotation tool called Tippanee, that is designed around the proposed anchoring approach; and provides its users with a collaborative environment where web users could help in improving the quality of textual content on the web by annotating, archiving, linking, sharing and semantically describing content on-the-fly.

CCS CONCEPTS

• **Information systems** → **Collaborative and social computing systems and tools**; **Crowdsourcing**; **Web interfaces**; *Social tagging*; Data extraction and integration; • **Human-centered computing** → **Web-based interaction**; • **Applied computing** → **Annotation**; • **Theory of computation** → *Pattern matching*.

KEYWORDS

Crowdsourcing, digital reference, edit distance, information exchange, online community, social web, tree matching, web annotation

ACM Reference Format:

Vishwajeet Pattanaik, Shweta Suran, and Dirk Draheim. 2019. Enabling Social Information Exchange via Dynamically Robust Annotations. In *The 21st International Conference on Information Integration and Web-based Applications & Services (iiWAS2019)*, December 2–4, 2019, Munich, Germany. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3366030.3366060>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

iiWAS2019, December 2–4, 2019, Munich, Germany

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-7179-7/19/12...\$15.00

<https://doi.org/10.1145/3366030.3366060>

1 INTRODUCTION

Since the inception of the World Wide Web (WWW) three decades ago, the Web has transformed from a mere medium for broadcast to a dynamic social environment, emerging as a pre-eminent mechanism for global communication, education, political-economic-cultural exchange and more [4]. This rising interest in the Web and its applications has led to a paradigm shift [19] in the ways we solve challenges (e.g., [16]), exchange knowledge/information (e.g., [13, 20]) and innovate (e.g., [12]); specifically by enabling Collective Intelligence through the Social Web [30].

Unfortunately, for a medium that has become so significant to our day-to-day lives, the Web has started to “wane in the face of a ‘nasty storm’ of issues” [32] such as clickbait, fake news, and misinformation [17]. This emerging trend of fake news is further fueled by the formation of so-called social media filter-bubbles [26], thereby causing ideological polarization [11] among social media users [2]. In addition to these issues, the Web’s expansiveness and ephemerality are also becoming increasingly worrisome [9, 28]. Studies have illustrated that the contents of most web pages now change within days, and the decay rate of web documents has dropped to nearly two years [25]. All of this is disconcerting, not only from end-users’ perspective but also for content creators, developers, scholars, and researchers; since these digital artifacts not only represent our web experience, but they also hold our collective knowledge [28].

As the literature suggests, a viable approach to tackle some of these challenges could be to leverage the ‘wisdom of the crowd’ [31, 36]. However, in order to design a multifaceted crowdsourcing information system [14] to save the Web, the system must fulfill three additional requirements. First, the system must act as “a conversation layer over the entire web”¹. Second, the system should treat web content as “first-class Web citizens” [8]; and finally, the algorithms through which the system interacts with web content shouldn’t be susceptible to the Web’s ephemerality. Interestingly enough, a class of systems that fulfill the first two requirements perfectly and has gained tremendous interest in recent WWW literature is “web annotation”. However, the state-of-art web annotation platforms (like *Hypothesis*) are either unable to cope up with ever-changing web content [1] or, explicitly rely on website owners/developers to be integrated onto the websites (in case of W3C’s Web Annotation Standards).

To address these challenges and to build upon our previous research [10, 27, 33], we set out to design an artifact that provides a twofold contribution. We first present a novel anchoring approach

¹*Hypothesis* article: “To enable a conversation over the world’s knowledge” | <https://web.hypothes.is/about/>

that can robustly reattach annotations on both static and arbitrarily generated web pages; and then explain how enabling users to interact with the Web by means of stable annotations could support end-user-oriented archiving, linking, sharing and describing (semantically) of web content, on-the-fly. With our work, we set out an alternative approach to achieve the vision of a “humanist format for re-usable documents and media” [24] by facilitating the amalgamation of state-of-web technologies with ‘wisdom of crowd’.

To evaluate the robustness of the proposed anchoring approach, we integrated the algorithm into a lightweight, user-friendly web annotation tool named Tippiancee². And, to validate our conjecture that crowdsourcing could help us resolve the aforementioned issues with the Web, we designed Tippiancee as a social web annotation platform where users can create, store, link, visualize and share textual annotations while browsing the Web. Additionally, we added features that allow Tippiancee users to add semantic descriptions to annotated contents, transclude annotations onto other web pages, and view orphaned annotations in their original state, even when the annotated content is altered or completely removed. As a preliminary evaluation of our research, we evaluated Tippiancee’s anchoring approach by annotating more than 650 random web pages and presented the tool to a small pool of end-users as part of our experiments.

2 RELATED WORK

Since our research is primarily related to web annotation tools and techniques, we first describe the state-of-art web-based text annotation tools. We then discuss the anchoring approach used by the Hypothesis web annotation tool. And, finally, describe a well-known pattern matching algorithm that exists at the center of our proposed anchoring algorithm.

2.1 Web Annotation Tools

Annotating, i.e., “the act of creating associations between distinct pieces of information” [29] has been recognized as a fundamental notion of hypertext systems since the inception of the WWW [21]. Due to its importance in the WWW community, several web annotation systems have been designed and deployed over the years, the most recent ones being Pundit, Genius and Hypothesis. Among these, *Hypothesis* has the most community support as it is a free and open-source platform. Based on the open-source JavaScript (JS) library Annotator.js, *Hypothesis* allows its users to add sentence-level annotations over web pages; and supports open critique and collaborative note-taking, by allowing users to highlight text, add/read and share annotations within private groups or in public.

Unlike the *Hypothesis* web annotation tool that is designed for end-users, the W3C’s web annotation recommendations are oriented towards website owners and developers. These recommendation include the Web Annotation *Data Model*, *Vocabulary* and *Protocol* and are successor of the W3C Open Annotation Data Model [29]. The model specification describes a structured format through which annotations can be shared and reused across different hardware and software platforms. The underlying data

model constitutes of three parts, namely: *a body*, *a target* and *a relation*.

While both the *Hypothesis* and W3C’s web annotation recommendations are widely considered the state-of-art in web annotations, however, both systems have specific drawbacks that prevent them from being used as crowdsourcing information systems for the Web. The *Hypothesis* annotation tool relies on a string-matching-based approach (called *Fuzzy Anchoring*) to reattach annotations. This means that change in annotated texts can cause incorrect reattachment of annotations or can orphan (i.e., when the annotation is no longer attachable) the annotations altogether. This is critical as frequent changes in the annotated content can render the anchoring approach useless. In an empirical study conducted by Aturban et al. [1], the authors found that 27% annotations on *Hypothesis* were already orphaned, and another 61% were at risk of being orphaned if the live web page changed. The drawback of the W3C’s web annotation recommendations, on the other hand, is that the choice to utilize these recommendations still lies with the website owners and developers, who are often bound by the technologies used in their platforms and may, therefore, choose not to adopt these recommendations.

2.2 Fuzzy Anchoring

To reattach its annotations *Hypothesis* utilizes a combination of multiple approaches relying on three selectors (namely, *RangeSelector*, *TextPositionSelector* and *TextQuoteSelector*) and four strategies (namely, *From Range Selector*, *From Position Selector*, *Context-first Fuzzy Matching* and *Selector-only Fuzzy Matching*)³. The first two strategies of the approach are based on XPath matching and are only usable in cases when there is no textual change in the document. Whereas, strategies 3 and 4 are useful when the underlying textual content is changed. These strategies utilize *approximate string matching* (fuzzy text search and comparison) algorithm, which is a combination of the Bitap matching algorithm [35] and Myers diff algorithm [22]. Since the approach is based on the ‘robust anchoring approach’ [3, 5], the selectors and the strategies of the approach rely entirely on ‘keyword anchoring’.

As mentioned previously, since the *Hypothesis* approach relies primarily on string or keyword matching, textual changes in annotated web pages can cause the annotations to be attached at incorrect locations. For example, we created an annotation “5,710,618” on the Wikipedia homepage on September 13, 2018 using *Hypothesis*. When we viewed the annotation again on November 4, 2018, we found that the annotation was now being reattached to “710 articles” (see Figure 1). And then again, when viewed on January 26, 2019 we found that the annotation was already orphaned. Similar examples of incorrectly attached *Hypothesis* annotations (as viewed on January 28, 2019) on the Wikipedia home page⁴ include (*annotated text* vs. *annotation reattached at text*):

- stars vs. seasons
- on, a league r vs. Hockey League for
- Albert Bridge vs. A left winger

²URL to Tippiancee’s source code, available on GitHub: <https://github.com/victor013/tippiancee-chrome-extension>

³*Hypothesis* article: ‘Fuzzy Anchoring’ | <https://web.hypothes.is/blog/fuzzy-anchoring/>

⁴*Hypothesis* annotations on Wikipedia main page: https://hyp.is/4ZuRcLJJEeiNkQvaWuwvng/en.wikipedia.org/wiki/Main_Page



Figure 1: (a) Annotated text as seen on Hypothesis Chrome extension vs. (b) the reattached annotation highlighted in blue



Figure 2: (a) Annotated text as seen on Tippiance Chrome extension vs. (b) the reattached annotation highlighted in yellow

We hypothesize that since the fuzzy anchoring approach relies so heavily on keywords, small annotations (i.e., maximum 3-4 words long) have a higher probability of being orphaned compared to long sentence-level annotations. However, further empirical studies would be required to validate the same. Incorrect reattachment of annotation (like the ones shown earlier) is unacceptable because similar changes to web pages could change the context of the annotation or could orphan the annotations altogether. While a few annotations might not be enough to evaluate the robustness of an annotation algorithm, however predicting when and how a textual web annotation would be orphaned can be a daunting task, as such predictions would depend on the half-life of the web page and the length of the annotated text. Continuing with our Wikipedia example “5,710,618”, we reproduced the same annotation onto our web annotation tool and found that our proposed anchoring approach was able to reattach the annotation successfully and at the correct location on the web page; even when the annotated content was changed (as shown in Figure 2).

2.3 Edit Distance

The Levenshtein Distance or, Edit Distance algorithm [18] is used to evaluate the difference between two sequences (a , b). The distance represents the number of single-character edits (insertions, deletions or substitutions) required to change one sequence into the other using the following equation:

$$\text{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \text{lev}_{a,b}(i-1,j) + 1 \\ \text{lev}_{a,b}(i,j-1) + 1 \\ \text{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise} \end{cases} \quad (1)$$

Similar to Levenshtein Distance, the Tree Edit Distance (TED) algorithm [37] is used to evaluate the similarity between ordered

labeled trees. Two trees are considered similar if their tree edit distance is below a predefined threshold depending upon the chosen path strategy. Although most available TED solutions are quite efficient, they cannot be used to compare HTML Document Object Model (DOM) trees because DOM elements are not necessarily labeled. We work around this problem by generating unique labels for annotated DOM elements. These unique labels are comprised of DOM attributes such as element name, id, class, etc. Once all nodes in a DOM element are uniquely labeled, they are then arranged in prefix notation. This information is then stored as anchor data. When reattaching annotations, web pages DOM elements undergo the same process. The achieved results are then compared to the stored anchor data using the Edit Distance approach.

3 SYSTEM OVERVIEW

3.1 Generating Anchors

To facilitate robust reattachment of annotations it is critical to uniquely identify annotated DOM elements. However, this cannot be done using only XPath, as changes in the document structure can alter the XPath of annotated element rendering the stored XPath invalid. To counter this challenge, Tippiance’s anchoring algorithm uniquely identify anchors by preserving the context and layout of the annotated elements. When a user annotates an element using the annotation tool, the system stores the DOM properties of the annotated element based on following strategy: (1) traverse the annotated DOM element sequentially (in prefix notation); (2) if the DOM element is a node, store its available properties i.e., *id*, *nodeName*, *className*, *alt*, *dataset*, *href* and *src*. And, if the DOM element is *#text*, store its *nodeName* and *nodeValue* and; (3) calculate the depth of each element and store it as *nodeDepth*.

The above-mentioned properties of the annotated elements are then stored as an array of JavaScript Object Notation (JSON) objects. Depending on the selected content, the system then uses the following strategies for anchoring different types of selections:



Figure 3: Illustration of anchored element (highlighted in blue) on Wikipedia homepage [above] and its HTML DOM as viewed on Chrome developer tool [below]

- *Annotating an element*: If the user annotates a complete DOM element, the anchoring algorithm selects the target DOM and transforms it into the above-mentioned JSON objects.
- *Annotating a #text node within an element*: If the user annotates some #text node within an element (but not all the text), the anchoring algorithm selects the target DOM and transforms it into a JSON object. However, in this case, it adds an extra boolean property 'annotated' to the selected #text node.
- *Annotating text in two or more elements*: If the user annotates texts from multiple DOM elements, the algorithm selects the common ancestor element as the target DOM and then generates its JSON object. Again, the 'annotated' property is added to the selected #text nodes.
- *Annotating substring within a #text node*: If the user annotates a substring from within a #text node, the system additionally stores the selected substring, and its starting and ending offsets within the #text node.

For instance, if a user selects the text “5,569,955 articles” on the Wikipedia main page (see Figure 3), the generated JSON object array would look like so,

```

[
  { id: "articlecount", nodeDepth: 0, nodeName: "DIV" },
  { href: "https://en.wikipedia.org/wiki/Special:Statistics",
    nodeDepth: 1, nodeName: "A" },
  { annotated: true, endOffset: 0, nodeDepth: 2,
    nodeName: "#text", nodeValue: "5,569,955", startOffset: 0 },
  { annotated: true, endOffset: 4, nodeDepth: 1,
    nodeName: "#text", nodeValue: " articles in ", startOffset: 0 },
  { href: "https://en.wikipedia.org/wiki/English_language",
    nodeDepth: 1, nodeName: "A" },
  { nodeDepth: 2, nodeName: "#text", nodeValue: "English" }
]

```

3.2 Reattaching Annotations

To reattach annotations, the algorithm searches for possible target elements using *getElement* DOM methods. The returned DOM elements are then compared to the anchor’s JSON object. The algorithm sequentially traverses the returned DOM element and its children. If the stored JSON object attributes match the attributes of the corresponding DOM element, it is considered a match. Whereas, elements that do not have the same attributes are considered mismatch.

In order to compare #text JSON objects to respective #text nodes, the algorithm uses approximate string matching. In the current version of Tippane, we use a JS implementation of approximate string-matching available on GitHub⁵. The JS program matches the two texts and scores their similarity with a value between 0 and 1, with 1 being a perfect match. The DOM element with maximum matches and minimum mismatches is selected as a viable target. If the *similarity index (SimIndex)*, i.e., the ratio between matches and anchor count of the identified elements exceeds the specified threshold (currently 0.5), the DOM element is considered as the final target for reattaching the annotation. In case no DOM element qualifies the minimum *SimIndex* threshold, the system assumes that the annotation is orphaned. Setting up a higher *SimIndex* threshold would mean that the system would only attach anchors that are perfect matches. Whereas, decreasing the threshold too low would cause selection of incorrect targets. Finally, if an annotation is orphaned, the annotation can still be reconstructed in its original form and can be viewed using Tippane’s *reconstruct anchor* feature.

While developing Tippane’s anchoring algorithm, we found that comparing DOM trees using conventional edit distance often leads to instances where multiple viable elements returned the same mismatch count. This led to us introduce a modified version of edit distance algorithm that analyses both mismatches and matches. Using both mismatches and matches, it becomes possible to identify a single element that has the maximum probability of being the target DOM element, making the approach more robust compared to *Hypothesis*’ approach. In Figure 4 we present the detailed process of reattaching annotations, illustrated as a flowchart.

⁵GitHub link to fuzzyset.js repository: <https://github.com/Glench/fuzzyset.js>

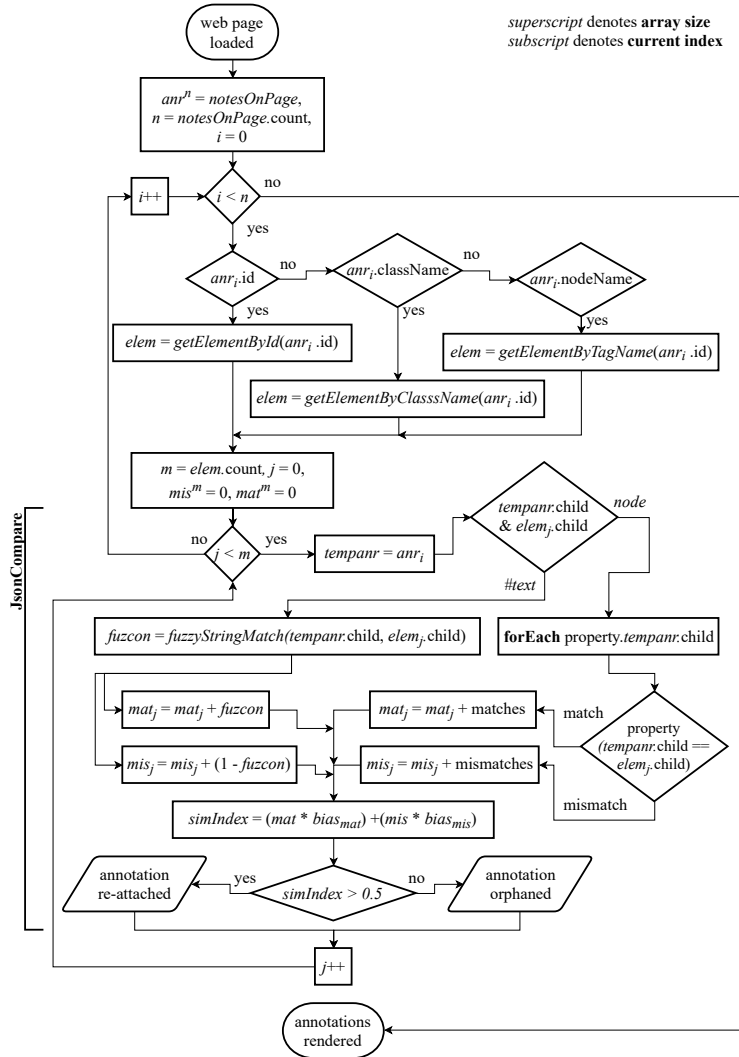


Figure 4: Tippanee’s anchoring approach presented as a flowchart

3.3 Implementation

The Tippanee annotation tool is available as a browser extension on Google Chrome Web Store. The extension is designed using HTML, CSS, JS and jQuery⁶, with additional JS libraries (vis.js⁷ for visualization and fuzzyset.js⁸ for fuzzy string matching). The Chrome extension supports both online and offline modes. When using

offline mode, user annotations are stored locally within Google Chrome’s local storage, and therefore the mode does not support annotation sharing. Whereas, in online mode, users can create and share annotations in groups, with the server-side code currently deployed on Google’s Firebase⁹. Irrespective of the modes, the computational processes for generating and reattaching anchors are always carried out exclusively on the client-side.

⁶jQuery homepage: <https://jquery.com/>

⁷GitHub link to visjs-network repository: <https://github.com/visjs/vis-network>

⁸GitHub link to fuzzyset.js repository: <https://github.com/Glench/fuzzyset.js>

⁹Google Firebase homepage: <https://firebase.google.com/>

Apart from the conventional annotation tool features, Tipanee also includes some novel components that we believe would allow the system to be utilized as a crowdsourcing information system for the Web. These features include:

- *Reconstructing anchors*: When transforming annotated texts into anchors, Tipanee’s anchoring approach stores partial DOM information of the annotated elements. This DOM information can then be utilized to reconstruct the annotations to its original state from when the annotation was first created. This is especially useful in the case of orphaned annotations, as the reconstructed anchor can help users better understand the relevance of the annotated content with respect to its surrounding text. Additionally, the same DOM information can also be utilized to support end-user-oriented information retrieval and web archival.
- *Linking and visualizing annotations*: This feature was inspired by the Linked Open Data Cloud visualization. The feature allows users to link created annotations and then visualize the same as a graph or ‘web of annotations’. Such graphs are an effective method of visually extending user-generated annotations and can facilitate meta-analysis and organization of ideas (e.g., [38]). The feature can also help enhance users’ web experience by presenting annotations and their associations in an easily understandable format.
- *Transclusions*: The system supports transclusions by allowing users to import and view annotations from other web pages onto the current web page. This is made possible by Tipanee’s transclusion server that uses the proposed anchoring approach to scrape requested web documents. After the annotated element is identified within the scraped document, the server pushes the DOM element onto the current web page. However, unlike conventional transclusion services, the pushed DOM element is only a part of the real-time copy of the requested web document.
- *Adding semantic descriptions*: As an added feature, Tipanee allows users to describe annotated text using Schema.org vocabulary. Users can currently only add, review, and change the stored semantic metadata, or they can use this metadata to search through stored annotations. However, in the future we would like to use this feature in support of full knowledge management life-cycle by allowing creation and exchange of new ontologies for organizations (e.g., [27]) and over the World Wide Web. Additionally, we would like to utilize user-generated semantic metadata for improving information access and search on the Web (e.g., [23]).
- *Similarity index*: The concept of SimIndex is a vital part of Tipanee’s annotation reattachment procedure. Apart from its role in the anchoring approach, the SimIndex of the attached annotations can also be exploited by end-users to track changes in annotated content. Users can track content changes on platforms such as news articles, Q&A portals, statistics and prices on e-commerce websites. In future, this feature could also support crowd-sourced fake news detection (e.g., [31]) and fact-checking (e.g., [7]).

4 EXPERIMENTAL EVALUATION

As part of the evaluation to validate the stability of the proposed algorithm, we created several web annotations using Tipanee over a duration of three months. To ensure that the conducted experiments are not biased, we replicated a random set of 735 *Hypothesis* annotations created by random users. We acquired these annotations using the ‘Annotation viewing and export’ tool¹⁰ made available by *Hypothesis* Labs. The tool allows users to view and export publicly shared *Hypothesis* annotations searchable by user, group, URL and tag. We manually replicated the selected annotations on Tipanee’s Chrome extension, by first browsing to the annotated web page; and then selecting and annotating the exact text as done in the respective *Hypothesis* annotations. During this process, we also found a few orphaned annotations, which we simply decided to ignore for this experiment. Once the 735 *Hypothesis* annotations were replicated in Tipanee, we left them aside for a month.

After a month, we revisited each of the annotated web pages and extracted the SimIndex(s) of the annotations using Tipanee. We were able to achieve a total of 675 SimIndex(s). Out of these, only 58 (8.59%) annotations had SimIndex(s) below 0.5, i.e., the annotations were now orphaned; while 611 (91.41%) annotations were successfully attached. Among the attached annotations 538 (88.0%) annotations had SimIndex(s) more than 0.9. Figure 6 presents a comparison of the number of annotations with respect to achieved similarity indexes (scaled from 0 - 1.0 to 0 - 100). The SimIndex threshold for the current version of Tipanee was set to 0.5, for this experiment. Our initial experiments clearly indicate a significant improvement of 12.41% over *Hypothesis*’ anchoring approach (which had 79% successful reattachments during Aturban et al.’s (2015) experiments); however, to demonstrate the robustness of our proposed approach in a reproducible manner, we plan to develop a first web annotation testbench that would simulate varying levels of change in a set of Alexa’s top-ranking websites¹¹. The said testbench would be crucial to decisively validate Tipanee’s anchoring approach, as it would allow us to mimic web page decay in real-world web pages in a controlled environment; without the need to wait for a live web page to change or decay.

4.1 User Evaluation

Additionally, to evaluate Tipanee’s features and to study user behavior when creating annotations on the Web, we presented our work to two different groups of users (12 web developers and 13 students). The groups comprised of both men and women between the ages 25 and 45; and were chosen in a way that they represented a mix of both technical and non-technical users. The groups were given a quick demonstration of Tipanee’s user interface and its features, after which they were asked to use the Chrome extension for a duration of at least 7 days. The users were requested to create and share annotations on web pages they visited regularly. After two weeks, the users were provided with a focused questionnaire enquiring about their experiences with the tool.

¹⁰ Annotation viewing and export tool: <https://jonudell.info/h/facet/>

¹¹ Alexa’s top-ranking websites: <https://www.alexa.com/topsites/category>

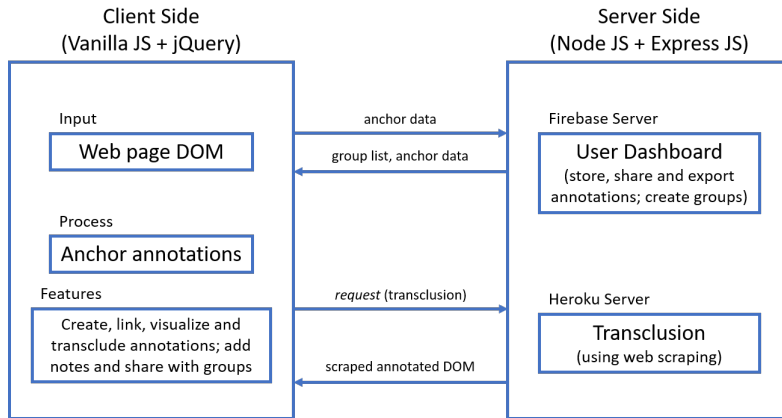


Figure 5: Illustration of Tippane’s system flow

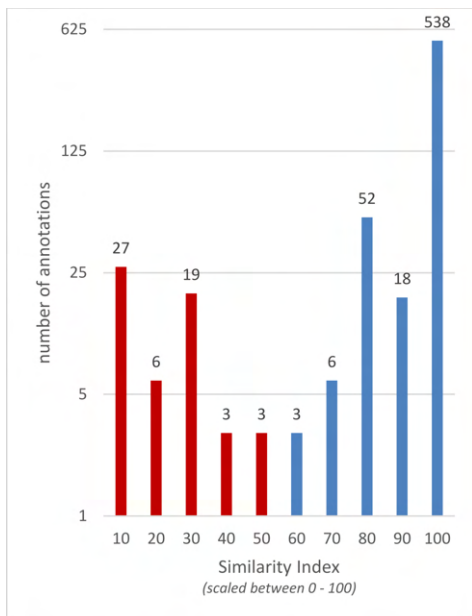


Figure 6: Similarity indexes of 675 annotations created on Tippane Chrome extension, scaled between 0 - 100; with orphaned annotations colored in orange and reattached annotations colored in blue.

As expected, most users suggested that they found the extension quite useful, especially for information retrieval and sharing. Interestingly, quite a few users suggested that they also used the tool for

social interactions and expression of opinions. Finally, users also reported that they created most annotations on educational websites, Q&A portals, news websites, and search engines. Although this preliminary study was carried out only for two weeks, it was valuable for our research as it provided us insights into how users interact on social-annotation platforms and helped us come up with elaborate ideas for future quantitative-quantitative evaluations.

5 DISCUSSION

In this section, we present the limitations of our work and elaborate on the legal aspects of web scraping, and its relevance with respect to our approach. We also briefly discuss the next steps of our work before we conclude the paper.

5.1 Limitations

We understand that the experiments conducted during the evaluation of Tippane might not be adequate to demonstrate the robustness of the proposed approach and that further evaluation is required for the same. However, since not all web pages are designed the same, neither is their page half-life; therefore it is difficult to determine which anchoring approach is the most robust. As a part of our future work, we plan on evaluating our proposed algorithm against the Fuzzy anchoring approach by creating a test bench with varying combinations of changes in both web page content and structure.

With respect to our formulae for calculating similarity indexes, the current threshold of 0.5 is only an estimated guess, and might not be ideal. Identifying the ideal threshold would require more extensive and long-term evaluations. Additionally, adding biases to the match-mismatch count of textual nodes could help enhance the accuracy of the algorithm; identifying the ideal values of these biases, is again a part of our future work.

5.2 Legal Aspects

Web scraping, caching and archiving has often been viewed as a severe ethical issue on the World Wide Web. Despite the fact that over the last couple of years the Open Data Movement has received tremendous support from governments, research institutes, laboratories, and libraries. Nevertheless, the idea of a truly open web seems to be a far-fetched notion. Several national libraries around the world have been supporting the concept of digital preservation and archiving, but most of these initiatives are hindered or affected by legal concerns, i.e., copyright infringement and piracy acts [34] (e.g., [6, 15]).

While designing the Tippianee platform, we studied these legal obligations in order to realize the scope of our system. Although web scraping, caching or archiving requires explicit permission from the web page owners, we argue that our system does not violate any piracy or copyright concerns, because of the following reasons: first, the system stores the context of the annotated content only to ensure that users do not lose their annotations and to improve the usability of orphaned anchors. In order words, content is saved primarily for the interest of the end-user. Second, the saved content is utilized solely for reattaching anchors and has no other commercial purpose whatsoever. Third, the context is stored exactly as it was published by the owner. And, the source of the content is clearly stated in the anchor's JSON object, meaning that the content is always linked to the source and the link is visible to the end-user. Lastly, enabling users to add semantic markup to annotations and creating links between annotations, helps in enhancing the user's web experience by encouraging semantic content creation and participation, therefore reinforcing and improving the social aspects of the Web.

6 FUTURE WORK AND CONCLUSION

The proposed anchoring approach provides a proof-of-concept for our hypothesis that using DOM information for generating anchors can facilitate more stable and robust annotations. Such DOM information can be especially useful to support end-user-oriented web archiving and information sharing processes. Finally, the social and semantic aspects of Tippianee can be utilized to support knowledge creation and sharing on the Web. As our next steps, we would like to optimize Tippianee's anchoring approach and add further features to support collaborative critiquing and knowledge sharing activities. And then, develop Tippianee into a lightweight, browser-based crowdsourcing information system that would allow users to contribute to web content by means of stable, interlinked and semantically rich annotations.

To conclude, in this paper, we have presented a novel DOM-oriented edit distance approach for enabling stable annotation and preservation of dynamically evolving web content. We presented evidence revealing that such a DOM-oriented approach could encourage smooth knowledge and information exchange over today's ephemeral web. And, we discussed how collaborative annotation tools like Tippianee could help tackle key challenges on the Web. Given the rising interest in collaborative platforms, annotation technologies and the importance of user's web experience, it is just a consequent step to team together all of these notions. Albeit the combination of these notions opens a wide design space for a whole

class of next-generation collaborative annotation platforms, the key challenge for making such systems a success is in providing robust anchoring mechanisms.

REFERENCES

- [1] Mohamed Aturban, Michael L. Nelson, and Michele C. Weigle. 2015. Quantifying Orphaned Annotations in Hypothesis. In *Research and Advanced Technology for Digital Libraries: 19th International Conference on Theory and Practice of Digital Libraries, TPDL 2015, Poznań, Poland, September 14–18, 2015, Proceedings*. Springer International Publishing, Cham, 15–27. https://doi.org/10.1007/978-3-319-24592-8_2
- [2] Vian Bakir and Andrew McStay. 2018. Fake News and The Economy of Emotions. *Digital Journalism* 6, 2 (2018), 154–175. <https://doi.org/10.1080/21670811.2017.1345645>
- [3] David M Bargeron, Alice Jane Bernheim Brush, and Anoop Gupta. 2010. Robust anchoring of annotations to content. US Patent 7,747,943.
- [4] Tim Berners-Lee. 2010. Long live the web. *Scientific American* 303, 6 (2010), 80–85.
- [5] A.J. Brush and David Bargeron. 2001. *Robustly Anchoring Annotations Using Keywords*. Technical Report. Microsoft. 19 pages. <https://www.microsoft.com/en-us/research/publication/robustly-anchoring-annotations-using-keywords/>
- [6] Jhonny Antonio Pabón Cadavid. 2014. Copyright Challenges of Legal Deposit and Web Archiving in the National Library of Singapore. *Alexandria* 25, 1-2 (2014), 1–19. <https://doi.org/10.7227/ALX.0017> arXiv:<https://doi.org/10.7227/ALX.0017>
- [7] Sylvie Cazalens, Philippe Lamarre, Julien Leblay, Ioana Manolescu, and Xavier Tannier. 2018. A Content Management Perspective on Fact-Checking. In *Companion Proceedings of the The Web Conference 2018 (WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 565–574. <https://doi.org/10.1145/3184558.3188727>
- [8] Paolo Ciccarese, Stian Soiland-Reyes, and Tim Clark. 2013. Web Annotation as a First-Class Object. *IEEE Internet Computing* 17, 6 (Nov 2013), 71–75. <https://doi.org/10.1109/MIC.2013.123>
- [9] Miguel Costa, Daniel Gomes, and Mário J. Silva. 2017. The evolution of web archiving. *International Journal on Digital Libraries* 18, 3 (01 Sep 2017), 191–205. <https://doi.org/10.1007/s00799-016-0171-9>
- [10] Dirk Draheim, Michael Felderer, and Viktor Pekar. 2014. Weaving Social Software Features into Enterprise Resource Planning Systems. In *Novel Methods and Technologies for Enterprise Information Systems*, Felix Piazzolo and Michael Felderer (Eds.). Springer International Publishing, Cham, 223–237. https://doi.org/10.1007/978-3-319-07055-1_18
- [11] Ivan Dylko, Igor Dolgov, William Hoffman, Nicholas Eckhart, Maria Molina, and Omar Aaziz. 2017. The dark side of technology: An experimental investigation of the influence of customizability technology on online political selective exposure. *Computers in Human Behavior* 73 (Aug 2017), 181–190. <https://doi.org/10.1016/j.chb.2017.03.031>
- [12] Bonabeau Eric. 2009. Decisions 2.0: the Power of Collective Intelligence. *MIT Sloan Management Review* 50, 2 (2009), 45. <http://proquest.umi.com/pqdlink?did=1625861001&Fmt=7&clientId=9268&RQT=309&VName=PQD>
- [13] Shixuan Fu, Gert-Jan de Vreede, Xusen Cheng, Isabella Seeber, Ronald Maier, and Barbara Weber. 2017. Convergence of Crowdsourcing Ideas : A Cognitive Load Perspective. In *Thirty eighth International Conference on Information Systems*. Association for Information Systems, 1–11. <http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1300&context=icis2017>
- [14] David Geiger, Michael Rosemann, Erwin Fieft, and Martin Schader. 2012. Crowdsourcing Information Systems-Definition, Typology, and Design. In *Thirty Third International Conference on Information Systems*. Association for Information Systems, 1–11. <https://ub-madoc.bib.uni-mannheim.de/32631/>
- [15] Lachlan Glanville. 2010. Web archiving: ethical and legal issues affecting programmes in Australia and the Netherlands. *The Australian Library Journal* 59, 3 (2010), 128–134. <https://doi.org/10.1080/00049670.2010.10735999> arXiv:<http://dx.doi.org/10.1080/00049670.2010.10735999>
- [16] Joshua Introne, Robert Laubacher, Gary Olson, and Thomas Malone. 2013. Solving Wicked Social Problems with Socio-computational Systems. *KI - Künstliche Intelligenz* 27, 1 (Feb 2013), 45–52. <https://doi.org/10.1007/s13218-012-0231-2>
- [17] David M. J. Lazer, Matthew A. Baum, Yoichi Benkler, Adam J. Berinsky, Kelly M. Greenhill, Filippo Menczer, Miriam J. Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, Michael Schudson, Steven A. Sloman, Cass R. Sunstein, Emily A. Thorson, Duncan J. Watts, and Jonathan L. Zittrain. 2018. The science of fake news. *Science* 359, 6380 (Mar 2018), 1094–1096. <https://doi.org/10.1126/science.aao2998>
- [18] V. I. Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady* 10 (Feb. 1966), 707. <http://adsabs.harvard.edu/abs/1966SPhd...10..707L>
- [19] Pierre Lévy. 2010. From social computing to reflexive collective intelligence: The IEML research program. *Information Sciences* 180, 1 (Jan 2010), 71–94. <https://doi.org/10.1016/j.ins.2009.08.001>

- [20] T.W. Malone (Ed.) and M.S. Bernstein (Ed.). 2015. *Handbook of Collective Intelligence*. MIT Press. 232 pages. <https://books.google.com/books?id=iR3iCgAAQBAJ>
- [21] Catherine C. Marshall. 1998. Toward an Ecology of Hypertext Annotation. In *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia: Links, Objects, Time and Space—structure in Hypermedia Systems: Links, Objects, Time and Space—structure in Hypermedia Systems (HYPERTEXT '98)*. ACM, New York, NY, USA, 40–49. <https://doi.org/10.1145/276627.276632>
- [22] Eugene W. Myers. 1986. An O(ND) difference algorithm and its variations. *Algorithmica* 1, 1-4 (1986), 251–266. <https://doi.org/10.1007/bf01840446>
- [23] Beate Navarro Bullock, Andreas Hotho, and Gerd Stumme. 2018. *Accessing Information with Tags: Search and Ranking*. Springer International Publishing, Cham, 310–343. https://doi.org/10.1007/978-3-319-90092-6_9
- [24] Theodor H Nelson. 2007. Transliteration: a humanist format for re-usable documents and media. *Transliteration.org* (2007).
- [25] Fatih Oguz and Wallace Koehler. 2016. URL decay at year 20: A research note. *Journal of the Association for Information Science and Technology* 67, 2 (2016), 477–479. <https://doi.org/10.1002/asi.23561>
- [26] Eli Pariser. 2011. *The Filter Bubble: What the Internet Is Hiding from You*. Penguin Group, UK.
- [27] Vishwajeet Pattanaik, Alex Norta, Michael Felderer, and Dirk Draheim. 2018. Systematic Support for Full Knowledge Management Lifecycle by Advanced Semantic Annotation Across Information System Boundaries. In *Information Systems in the Big Data Era*, Jan Mendling and Haralambos Mouratidis (Eds.). Springer International Publishing, Cham, 66–73. https://doi.org/10.1007/978-3-319-92901-9_7
- [28] Jeffrey M. Perkel. 2015. The trouble with reference rot. *Nature* 521, 7550 (May 2015), 111–112. <https://doi.org/10.1038/521111a>
- [29] Robert Sanderson, Paolo Ciccarese, and Herbert Van de Sompel. 2013. Designing the W3C Open Annotation Data Model. In *Proceedings of the 5th Annual ACM Web Science Conference (WebSci '13)*. ACM, New York, NY, USA, 366–375. <https://doi.org/10.1145/2464464.2464474>
- [30] Detlef Schoder, Peter A. Gloor, and Panagiotis Takis Metaxas. 2013. Social Media and Collective Intelligence—Ongoing and Future Research Streams. *KI - Künstliche Intelligenz* 27, 1 (Feb 2013), 9–15. <https://doi.org/10.1007/s13218-012-0228-x>
- [31] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake News Detection on Social Media: A Data Mining Perspective. *SIGKDD Explor. Newsl.* 19, 1 (Sept. 2017), 22–36. <https://doi.org/10.1145/3137597.3137600>
- [32] Olivia Solon. 2017. Tim Berners-Lee on the future of the web: 'The system is failing'. , 2017 pages. <http://www.nytimes.com/2010/04/27/world/27powerpoint.html>
- [33] Shweta Suran, Vishwajeet Pattanaik, Sadok Ben Yahia, and Dirk Draheim. 2019. Exploratory Analysis of Collective Intelligence Projects Developed Within the EU-Horizon 2020 Framework. In *Computational Collective Intelligence*, Ngoc Thanh Nguyen, Richard Chbeir, Ernesto Exposito, Philippe Anrioté, and Bogdan Trawiński (Eds.). Springer International Publishing, Cham, 285–296. https://doi.org/10.1007/978-3-030-28374-2_25
- [34] Richard S. Vermut. 1996-1997. File Caching on the Internet: Technical Infringement or Safeguard for Efficient Network Operation. *Journal of Intellectual Property Law* 4 (1996-1997), 273.
- [35] Sun Wu and Udi Manber. 1992. Fast Text Searching: Allowing Errors. *Commun. ACM* 35, 10 (Oct. 1992), 83–91. <https://doi.org/10.1145/135239.135244>
- [36] Amy X. Zhang and Justin Cranshaw. 2018. Making Sense of Group Chat Through Collaborative Tagging and Summarization. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 196 (Nov. 2018), 27 pages. <https://doi.org/10.1145/3274465>
- [37] K. Zhang and D. Shasha. 1989. Simple Fast Algorithms for the Editing Distance Between Trees and Related Problems. *SIAM J. Comput.* 18, 6 (Dec. 1989), 1245–1262. <https://doi.org/10.1137/0218082>
- [38] J. Zhao, M. Glueck, S. Breslav, F. Chevalier, and A. Khan. 2017. Annotation Graphs: A Graph-Based Visualization for Meta-Analysis of Data Based on User-Authored Annotations. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (Jan 2017), 261–270. <https://doi.org/10.1109/TVCG.2016.2598543>

Appendix 3

III

V. Pattanaik, I. Sharvadze, and D. Draheim. Framework for peer-to-peer data sharing over web browsers. In T. K. Dang, J. Küng, M. Takizawa, and S. H. Bui, editors, *Future Data and Security Engineering*, pages 207–225, Cham, 2019. Springer International Publishing



Framework for Peer-to-Peer Data Sharing over Web Browsers

Vishwajeet Pattanaik²(✉) , Ioane Sharvadze¹(✉), and Dirk Draheim² 

¹ Microsoft, Tallinn, Estonia
ioane.sharvadze@gmail.com

² Information Systems Group, Tallinn University of Technology, Tallinn, Estonia
{vishwajeet.pattanaik,dirk.draheim}@taltech.ee

Abstract. The Web was originally designed to be a decentralized environment where everybody could share a common information space to communicate and share information. However, over the last decade, the Web has become increasingly centralized. This has led to serious concerns about data ownership and misuse of personal data. While there are several approaches to solve these problems, none of them provides a simple and extendable solution. To this end, in this paper, we present an application-independent, browser-based framework for sharing data between applications over peer-to-peer networks. The framework aims to empower end-users with complete data ownership, by allowing them to store shareable web content locally, and by enabling content sharing without the risk of data theft or monitoring. We present the functional requirements, implementation details, security aspects, and limitations of the proposed framework. And finally, discuss the challenges that we encountered while designing the framework; especially, why it is difficult to create a server-less application for the Web.

Keywords: Data ownership · Decentralization · Human-computer interaction · Peer-to-peer · Social web · Security · Web apps · WebRTC

1 Introduction

The World Wide Web was originally designed to be a decentralized network and ‘a common information space’ where users could ‘communicate by sharing information’¹. However, as pointed out by numerous researchers, over the last decade the Web has become increasingly centralized [13, 17, 20]. Furthermore, the rising interest in social web platforms like Facebook, Twitter, and Google [9] is making the situation worse; as more and more users are drawn towards such closed platforms. Such platforms that have already opted to centralize user resources into closed data silos [17], each controlling their silos using proprietary authentications and access control mechanisms.

¹ W3C—The World Wide Web: A very short personal history.

Due to this nature of web applications, users typically end up creating dedicated accounts on multiple platforms; thereby bounding themselves to particular services and resources [17]. Furthermore, users trust such web applications and service providers to store and manage their personal data with the intention to receive personalized services. However, as recent incidents [2] indicate such centralized data silos could also be utilized to harvest user data [8], manipulate user mindset [1], and to spread fake news and propaganda [12].

Events like these have given rise to some serious concerns, not only about data ownership and misuse of personal data; but also about the inability of such applications to allow for secure data exchange between different platforms. While researchers have developed several remarkable artifacts [3, 4, 16–18] that attempt to resolve these challenges, however, most of these platforms are still not being utilized by end-users as much. A possible reason for this lack of acceptance might simply be the lack of technical know-how and coding skills required to fully utilize such artifacts. It's difficult to determine precisely 'why end-users might decide to adopt a technology and not the others'; as this (i.e., Acceptability Engineering) is still a rather new field of research, which needs to be developed further [10]. We argue that in case of general internet users, the reasons might simply be the lack of familiarity [11], and the heterogeneity of platforms and tools available online. We are convinced that by providing end-users with a secure platform that can be integrated into any browser, we should be able to provide end-users with a more fluid experience when sharing data/information online.

To this end, we set out to provide a solution that attempts to reduce dependency on centralized servers (i.e., data silos) and empowers end-users with true data ownership. We present a platform-independent, browser-based framework for sharing data between applications over peer-to-peer (P2P) networks. The framework aims to empower end-users with complete data ownership, by allowing them to locally store shareable data and then share it directly with other users; without the risk of data theft or monitoring.

Ultimately, we would like to integrate the said framework into our ongoing research [5, 14] on 're-decentralizing the Web'; by eliminating the need for servers and allowing users to communicate with each other without any middleware. The proposed framework is designed to enable P2P communication for a crowdsourcing information system (CIS) called **Tippanee** [14]. The Tippanee CIS is currently being developed as Google Chrome browser extension that allows end-users to weave, critique and share web page elements, independent of content ownership on the Web. However, as mentioned earlier, due to the recent increase of interest in user privacy and data-ownership, it has become imperative that such a P2P framework must be designed as a 'generic system'. Developing such a generic framework should empower both web-developers and end-users, by providing Application Programming Interface (API) based server-free communication between applications and by preventing misuse of user data, respectively.

In this paper, we present the client library and web services, that allow applications to store data on user devices and share them over a P2P network.

We explain the functional requirements, implementation details, security aspects, and limitations of the framework. And finally, we discuss the challenges that we encountered while designing the framework; and especially, why it is difficult to create a server-less application for the Web.

2 Related Work

In this section, we briefly summarize recent scientific contributions that attempt to tackle the issues of data ownership and privacy. Each of these platforms allows users to create and share data via decentralized networks. Analyzing these platforms and understanding their approaches, contributions and limitations helped us identify the gaps; and thus, derive a clear problem definition.

2.1 Musubi

Musubi [4] is a mobile social application platform that enables users to share data in real-time feeds. The platform ensures data safety and privacy by supporting end-to-end public key encryption and allows users to interact with their friends directly through their address books. Additionally, the platform provides end-users with complete data ownership by allowing them to store all their data on their phones. The framework is limited to mobile app communications; however, since establishing direct P2P connections over mobile networks (3G) is not possible, data transfer is completely dependent on a centralized service Trusted Group Communication Protocol [4].

While Musubi's goals might seem similar to ours, there are some subtle differences in both. For instance, unlike the proposed system, Musubi only supports group sharing and does not support public data sharing. Also, Musubi always requires a server for data transfer, i.e. every single message sent between users must be routed through a server. Contrary to this, the proposed system does not require a server to store or relay data (in case both sender and receiver are online).

2.2 CIMBA

CIMBA or Client-Integrated Micro-Blogging Architecture [17] is a decentralized social web application that enables data ownership by allowing users to choose where their data is stored. The architecture uses WebID² [7, 19] and WebID-TLS³ to identify users at the Web scale and to authenticate requests.

The platform fully decouples the application web server from the user's database. And thus, allows users to have full control of their databases. The platform allows its users to decide which data they would like to share, with which web application, and which data they would like to keep private. As an added advantage, the platform also supports data reuse by allowing web apps to reuse the user's social graph; thus further unlocking the silos [17].

² W3C—WebIDs and the WebID Protocol.

³ W3C—WebID Authentication over TLS (editor's draft).

2.3 Solid

Solid [13, 18] is a decentralized platform for social web applications, that allows users to manage their data independent of web applications. Unlike in conventional web applications where user data is stored and managed by the web application provider, Solid users are required to store their data in a personal online data stores (called pods). Web applications are allowed to access users' data, based on the permissions provided by the users. The users are identified using WebID [7, 19] and have full control over how their data is accessed. Users can also switch between applications and pods at any time. The platform uses Resource Description Framework (RDF) based resources to exchange data between applications and pods [13, 18].

We argue that although Solid supports data ownership, however, it still stores data on non-user devices. Also, for an average non-technical users tasks like setting up servers or finding free hosting services might seem really tedious. We are of the opinion that, removing the need for any configuration and storing users' data onto their personal computers should improve the users' experience. Also, since linked-data does not provide solutions for real-time P2P data sharing, it would be challenging to develop real-time social applications such as chats on such a platform.

2.4 Dokieli

Dokieli [3] is a decentralized browser-based authoring and annotation platform. Similar to CIMBA and Solid, the platform supports social interactions and allows users to retain the ownership of their data. Documents created in dokieli are both independent and interoperable, as they follow the standards and best practices of HTML, RDF, and Linked Data [3]. Like CIMBA and Solid, dokieli too enhances user's data ownership experience. Unfortunately, all of these platforms are bound by their technologies; which can discourage developers from adopting these recommendations. Finally, as mentioned earlier, none of these platforms allow storage on personal computers and end-to-end communication.

3 Challenges

Given this lack of a generic, application-independent data-sharing platform for P2P networks over the Web; in this paper, we attempt to provide answers to the following questions:

- **How to establish P2P network between web browsers?**

The aim of this research question is to understand how P2P networks could be established over web browsers; and to examine how [Socket.IO](#) and [WebRTC](#) could be used to establish communication in such a scenario? This is also important as answering this question would help solve the NAT traversal [6] problem using JavaScript.

- **How to create a server-less P2P network?**

This question is critical, as the answer would help us establish whether P2P networks can be initiated in truly server-less conditions. And, if that's not possible; what other alternate approaches could be used to reduce dependency on servers?

- **How to forward messages to a user who is currently offline?**

Since the proposed framework has to be deployed over P2P networks, it is vital to understand how the system would behave if the message receiver is not online. By brief analysis, it would mean that one would not be able to establish a P2P connection in such scenario. However, would it be possible to devise a fallback solution that could still deliver messages? If so, how would the system function?

3.1 Functional Requirements

As stated previously, since the proposed framework was initially designed to be a part of the Tippianee platform; to model adequate solutions, we decided to first examine the requirements from our previous work in data management [5, 14] and then generalize these requirements for other applications. Since Tippianee is a social application, it needs the ability to share and control data visibility. Having such restrictions on data access means that proposed framework should be built with security and privacy in mind. Also, from a developer's perspective, it should be easy to integrate this functionality, and so the framework should be simple enough to hide data management complexity from application developers.

Keeping these factors in mind, we laid out the following functional requirements for the proposed P2P framework:

- **Private Share** means that the web content stored by a user (hereby referred to as *sender*) should be viewed/modified only by the same user and the data should only be available on the user's local computer.
- **Public Share** means that all members of *sender's* community (or, network) should be able to search and download the stored web content.
- **Private Share Between Friends** is a mix of *Private Share* and *Public Share*, such that in this case the *sender* can share the stored web content with a selected set of members within the network. So, while for the selected members of the network, the content (or, message) would seem public; for the other members, the content would seem private. This means the framework should handle the delivery of the data to the peers from the *sender's* device.
- **Offline Peer.** This requirement is crucial in the P2P framework as members of the network might not always be online to receive messages. Hence, the framework should be able to securely hold the message until the *receiver* comes online.
- **Saving the Data.** The framework should allow local storage, so users can view stored content even when they are not connected to the network; or in-case the shared content is lost or deleted by the *sender*.

- **Security & Integrity.** While the proposed network would be distributed, the framework should pay key attention to data authenticity. This is imperative as malicious users should not be allowed to tamper with the content shared by the *senders*. And the *receivers* should be able to verify if a message was sent by an authentic user. To enhance data ownership further, the servers in between should not be able to extract or read the shared data; i.e., the stored data should be encrypted.
- **Technical Requirements.** Finally, the framework should work with browser-based web applications and browser extensions. Since web applications and extensions may present different technical requirements for framework, the framework should be completely generic. Also, the framework should be able to run as background page (within the browser), so that the application can receive connections and share/receive data even when the extension is closed.

3.2 Limitations

During the initial literature review, we found that it is not possible to establish a P2P communication between users without a third party server [6]. The reason for this is that, in the real world environment, most devices on the Web are hidden behind Network Address Translators (NAT). This means that not all the users on the Web have a unique Internet Protocol (IP) address. NATs provide devices with a local address, which is only unique within the local network and not within the Wide Web. It is the NAT's job to translate local IPs into unique public IP and port configurations for communication with outside systems. This means that if a device is connected to a network with NAT, multiple devices within the network will receive same public IP, but with different ports configurations. Once a device is momentarily disconnected from the network, it might receive a different IP and port configuration; thereby forbidding incoming connection requests to the device.

To overcome this problem, peers should start requesting connections to each other simultaneously. In such a case, the NAT will most likely (in 64% cases for TCP connections) enable Peer-to-Peer connection. This technique is referred to as Hole Punching [6]. Unfortunately, this also means that in some 36% cases, it would not be even possible to send data to a peer without a middleware server. Keeping this issue in mind, one must provide a fallback mechanism in the form of a relay service; which we describe further, in the following sections.

4 Proposed Framework

Building on the requirements and limitations described in the previous sections; in this section, we present the different design choices we had to keep in mind while designing the proposed framework.

4.1 Why Server?

P2P Connection Establishment. In order to establish a P2P connection, it is necessary for both peers to share their IP addresses with one another. When both peers have each other's IP addresses, they will need to request connections simultaneously. However, as mentioned earlier, since most devices exist behind NATs, establishing a connection when either peer is behind a NAT would require NAT hole punching. For symmetric NAT traversal, the devices would have to start sending data to peer public IP & port; for this reason alone it is imperative to have a server, that allows two peers to share their public IP addresses and thus assist in establishing the P2P connection.

This challenge was resolved by the advent of Google's WebRTC (web real-time communications) protocol. WebRTC is a set of APIs that is implemented by most contemporary browsers. It was first implemented in Google Chrome browser and thus it fulfills the browser supporting requirements of our framework. Since WebRTC is only an API, to provide a complete solution, we had to develop a signaling implementation that could be used to establish WebRTC connection. To this end, we decided to use Socket.IO [10], since the library is well documented and widely popular.

With the WebRTC approach, before establishing the connection, both peers would have to first connect to the signaling server. Once both peers have exchanged their IPs, they can connect with each other and start sharing data directly (without the server).

Public Data Holding. In order to forward publicly shared data (or, messages), the data would have to be stored on a public server. Also, since members of the network might not be familiar with one another, it would not be possible to establish a P2P connection. And sending a request (to store the shared public message) to all members of the network would be extremely inefficient. Hence for the proposed network, we decided that all publicly shared data would be stored and indexed in the server. Peers who might choose to access the public data would be allowed to request for the same.

Sharing Data When Peer Is Offline. Another reason for using a server within the framework is that a peer might not be online when some data is being shared. Imagine a scenario where the *sender* shares a message with the *receiver*, but the *receiver* is not online. In this case when the *receiver* comes online, the *sender* may or may not be online. Without a server in place, it would not be possible for them to share the data unless both peers are online simultaneously. Hence, by simply using a server to hold undelivered messages temporarily, the issue of sharing messages with offline peers could be resolved.

Keeping these factors in mind we decided to include a server (as a fallback mechanism) in the proposed P2P framework.

4.2 Server Architecture

Taking into account the challenges we described in Sect. 3; in order to empower users with privacy and data ownership, the said server must achieve the following goals: the server should not be application-specific, the server’s APIs should be implemented separately, and the server’s role in the message exchange process should be kept to a bare minimum. Ultimately, the said server should be application-independent and should have negligible access to the hosted data.

Decoupling the applications from the server implementation in this manner would allow the use of different servers with different implementations that support the same APIs. This will provide server maintainers the freedom to choose how much (undelivered) data to hold, which applications to communicate with and to add additional security checks to secure the server. By nature of this solution, since the server would have no direct access to the stored data, it would not be able to check data integrity; and hence, users would be in charge of validating sender identity and data integrity.

Interestingly, the Musubi platform [4] also uses a similar methodology to secure user data. The platform allows developers to use common servers to share data between users. However, the data is encrypted such that only the users themselves can verify the sender’s identity and data integrity. However, unlike the proposed solution, Musubi is designed for mobile phones and lacks peer-to-peer data sharing support.

Solution. To accomplish the requirements of P2P data sharing, we designed two different services. The first service allows peers to establish a connection (which we refer to as “Live-Rooms”). And, the second service supports message exchange and peer communication with offline users. We refer to this service as the “Message Box”. As illustrated in Fig. 1, the first service is for establishing P2P connections, while, the second is just temporary data storage space, similar to a real-world post office. Both services are designed to be independent, and thus applications can choose different service providers, or opt to support only one of them.

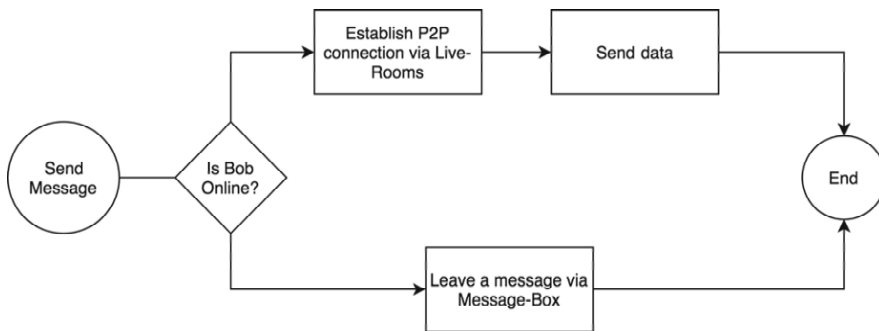


Fig. 1. Illustration of the message sending process

4.3 Live-Rooms

The ‘Live-Rooms’ service is designed to enable fast (real-time) data sharing between users with P2P traffic. The service is designed as a common space where online users gather and wait for incoming connections. Each user can connect to the ‘Room’ associated with their user ID. Whenever a user wants to connect to other users, they can connect to the Live-Room and ask for connection to specified users’ ID. Once the request is complete, users can then decide whether they want to accept an incoming connection or not. Once two users are connected, they can send/receive data and messages, verify delivery and finally close connections when needed.

4.4 Message Box

The ‘Message Box’ service has a more extensive role within the framework. In order to achieve the required goals, the service must fulfill several functionalities: (1) list message-ID’s for user; (2) download messages by message-IDs; (3) save shared messages; (4) list public messages by keys; (5) download public messages by keys; and (6) save public messages with keys.

Among these, the first three functionalities are required for message relay (when the *receiver* is not online). Whereas, the last three functions are required for storing/querying public data shared by users. Based on these requirements and functionalities, for our current implementation, we used a popular client-server architecture called REST [15].

4.5 Client-Side

To facilitate the development of various applications based on the proposed framework, the complexity of data handling is hidden from the application developers. The library is compiled into a single JavaScript file so that it can be added to HTML as a single script. Doing so allows the developers to use the framework’s functionalities with a simple API. Also, this prevents the client library from exposing the inner implementation to the users, while providing an abstraction to manage (save, delete, receive, query) public, private and shared data.

Configurations. For the initial version of the client library we provide five configurable (optional) parameters including: ‘Live-Rooms’ URL, ‘Message-Box’ URL, ‘Message-Box’ synchronization interval, Local Database Name, and ‘Live-Rooms’ WebRTC configuration’ (that contains STUN server URLs). For ‘Live-Rooms’ and ‘Message Box’ URLs, the default public service URLs are used. In our current implementation, these default services are maintained on the [Heroku Cloud Application platform](#). In our implementation, the local database names are generated with default names, which are generated based on user IDs. This feature is important as, the database name can be changed based

on changes in user ID; hence preventing users from reading other users' data. In case of automatic database names, the application uses the corresponding user's database.

Also, to further assist in establishing a P2P connection, 'Live-Rooms' WebRTC configuration is designed to allow changes to the default STUN server URLs. This provides developers with some flexibility as they can decide whether they would like to use the default STUN server (maintained by Google) or their own STUN servers (for more control over performance and security).

API Usage. To use the proposed library in a web application, developers simply have to import a single JavaScript library file. The library includes the following Javascript methods:

- **sync()** The Sync method provides a way to force synchronization between 'Message Box' and application. This is especially useful if the application needs to get fresh data and cannot wait for the scheduled update interval. In applications, such a scenario might occur when the user forces synchronization by clicking the refresh button.
- **fetchPublicDataByKey()** method comes into play when a user sends a request to read the public data. The client library fetches the 'Message Box' for all available public data and returns it to the user. It is important to note that, the returned public data is not saved onto the user's device unless the user requests for the same. Users are allowed to view the data available on the public channel, and they may then decide whether they would like to store a copy of the public data on their local device.
- **publish(key, callback)** The Publish method simply makes the shared data publicly available. This requires a key to make the data identifiable by other users. Once a piece of data is published, a copy of the same is then stored onto the server (by the 'Message-Box' service).
- **getByKey(key, callback)** This method is used to retrieve data with an application-defined key. Note that the key is not required to be unique, so users would either receive a list of results in the callback or, null if nothing is found.
- **getByAuthor(key, callback)** This method, works similar to the **getByKey** method, however the results are queried by author ID (i.e., the author's public key).
- **saveData(data, sharedWith, callback)** This method stores the shared data onto to server, but only until the recipient of the message comes online.
- Finally, the **listenDataChanges(callback)** method is a convenience method that synchronizes the application with the storage. If a user sends a message via 'Live-Rooms' or, retrieves a message through the 'Message-Box', the user gets new data objects in the callback. The method also allows for local database querying and displays fresh results if needed.

Background Synchronization. The proposed library is designed to work with both web applications and with browser extensions (only Google Chrome for now). The difference being that extensions can store unlimited data (within browser storage) and have the ability to synchronize data even if the user has not opened the application.

In the Google Chrome browser, extensions have a notion of [background pages](#); that allow application developers to run scripts even when the HTML page is not open (or selected) by the user. The library uses the background page to establish and maintain ‘Live-Rooms’ connections, and to share/receive data by fetching new messages from the ‘Message Box’ at scheduled times. Unlike browser extensions, regular web applications can only support background synchronization when the application is open in a browser tab.

5 Implementations

As mentioned in Sect. 4.2 we designed the ‘Live-Rooms’, ‘Message Box’ and ‘Client Library’ modules as distinct entities. This helped us reduce coupling and allowed us to extend the functionalities without modifying other modules. The implementation code of the modules is available as a [repository](#) on GitHub.

5.1 Live-Rooms

The ‘Live-Rooms’ is designed using the Socket.IO library. The module is divided into two separate parts: the Web-Client and the Server-Client. The server-client is written in JavaScript Node.js framework. While the module could be designed using other JavaScript web frameworks, we choose Node.js as it is the most popular choice among developers⁴.

The server is implemented using the Socket.IO library, as, for the proposed framework both server and client need to send bidirectional events. Consider the scenario, where a client might need to send several messages and then wait for an event from a server. For this reason, the client needs to have a consistent connection with the server, so that the server can notify the client when other peers request connection. Other solutions like pooling could be applied in this scenario, but since clients will have to send several signaling messages through the server, managing several signaling messages with pooling approach would be less efficient and complex. The socket implementation, on the other hand, can tackle multiple bidirectional messages, due to its notion of events. The first (“connect”) event occurs when the user connects to the server. At this point, the user has a ready socket that can be used to send and receive a message to and from the server.

⁴ [Stack Overflow—Developer Survey Results 2017](#).

Message Contract and Events. Once a user is connected to the server, the server expects several types of events from the user. Every message sent by the user must be a JavaScript object with *fromPublicKey*, *toPublicKey* and *data* attributes. The ‘fromPublicKey’ holds the public key of the message sender, the ‘toPublicKey’ holds the public key of the receiver, and other attributes are shared as ‘data’.

In our implementation, the first event expected by the server is called ‘enter_my_room’. The event fires when a user is ready to provide his/her public key. The server then adds the user to a ‘room’ named with same public key. In Socket.IO, this notation of ‘room’ enables us to label sockets. This also helps to check if a peer is online and if the peer has received redirect messages from other peers; thereby enabling multiple parallel signaling between peers. If an event fails, the server usually sends an “error” event to notify the client about the failed action. The server also sends a failed message so that the client can reset their state and try again if needed.

Once a user joins the said room, the server is ready to notify all peers of the connection requests. The user can also initiate a peer connection at this stage. To initiate connection requests, the event ‘connection_request’ is called. Note that this event also requires peer public key, so that peers can receive an event. At this stage a peer can either accept or ignore ‘connection_request’. By accepting the request client starts sending events named ‘signalling_message’; which are redirected from one peer to another. The process of sending WebRTC signaling messages leads to establishment of P2P connections (as illustrated in Fig. 2).

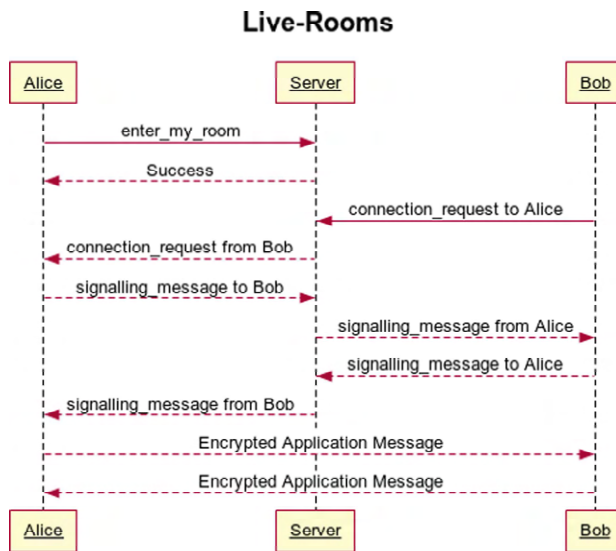


Fig. 2. Illustration of the P2P network establishment process

5.2 Message-Box and Storage

The Message-Box is designed using the JavaScript framework called [Express.js](#). The Message-Box is designed as a REST API and includes methods to – (1) list all messages for user, (2) get messages by message IDs, (3) save messages, (4) list all public messages by keys, (5) get public messages by IDs, and (6) publish messages. Message-Box documentation is deployed on online API documentation tool (see Fig. 3), that provides the possibility to describe API easily, add sample responses, and mock the functionalities. For storing messages, we used the NoSQL database Mongo DB (please see Fig. 4 for the message schema.) in our implementation.

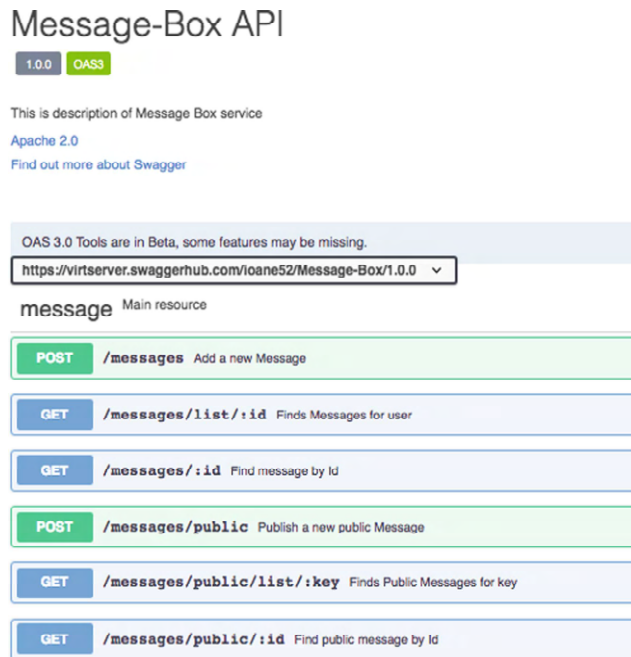


Fig. 3. Screenshot of the Message-Box API

Deployment and Evaluation. To evaluate the proposed framework and the implemented artifact, we deployed a web application on Heroku with the database established in [mLab](#). The mLab database platform was then connected to the ‘Message-Box’ service and the Heroku server.

DataController. The DataController is the main class that handles all data within the system. It holds the system state and delegates functionalities to


```

var messageSchema = mongoose.Schema({
  message: String,
  sharedWith: [String],
  key: String,
  public: Boolean,
  author: String
});

```

Fig. 4. Message database schema

the different data controllers, namely: Local, Live and Cloud. The Local DataController's task is to save/query data in local storage. While the Live DataController interacts with 'Live-Rooms' service and saves/receives data via P2P connection, it hides all the logic of peer connection and data retrieval; the Cloud DataController interacts with 'Message Box'. The DataController class unites all of these types and hides the functionalities of each of these DataControllers from the others (as illustrated in Fig. 5). The DataController also handles configuration of different services and passes corresponding parameters when needed. Internally, the DataController also handles unique ID creation, so that library users don't have to create IDs for messages.

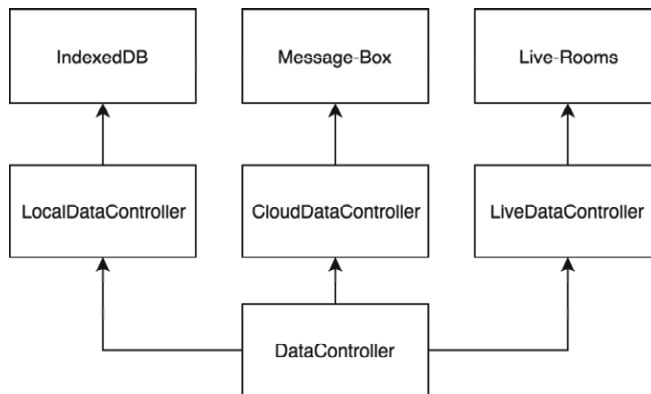


Fig. 5. Structure of DataController

The different stages of the DataController are as flows:

- When initialized, it creates all the controllers.
- During synchronization it tries to fetch data from Cloud DataController.
- The 'getByKey' and 'getByAuthor' function delegates to Local DataController, if the user is searching current data.
- The 'saveData' first saves data with Local DataController, then tries Live DataController, if a peer is not online, the message is saved to the Cloud DataController.
- The 'listenDataChanges' waits for data updates from Live and Cloud DataControllers.

Live DataController. The Live DataController as described above, is responsible for using ‘Live-Rooms’ service. Upon creation, it connects users to their rooms and listens to ‘connection_request’ and ‘signaling_message’ events. When a ‘connection_request’ is received, it starts the P2P connection establishment process. The connection establishment process starts with the gathering of ICE (Interactive Connectivity Establishment) candidates. The ICE candidates are then sent to the remote peer using ‘signaling_message’. The ICE candidates are needed to perform NAT traversal. By default, ICE candidates are configured to be free STUN services provided by Google. When remote peers receive a signaling message, they also start to gather ICE candidates and send them to their peer using ‘Live-Rooms’. After the ICE candidates are shared, data channels are opened and users can start sending data using the P2P connection. The Live DataController also holds the opened data channels so that it doesn’t have to create new connections every time user sends a piece of information. When the data channel is broken, new connection establishment process is started; if multiple errors occur, a return null callback is sent to the sender, so that other service can try sending information.

Cloud DataController. Cloud DataController handles ‘Message-Box’ service interaction. It has several public functions: ‘sync’, ‘save’, ‘publish’, and ‘fetch public by key’. When a ‘sync message is fired’, it connects to the message box listing endpoint, gets a list of messages and downloads them one by one. ‘Save’ and ‘publish’ functions on the other hand, only send a message to the server. ‘Save’ requires a list of public keys, that have access to the message. Whereas, ‘publish’ makes the message publicly available and associates them with the key. Finally, the ‘fetch public by key’ searches for public messages associated with key and downloads them.

Local DataController. For implementing Local DataController [IndexedDB](#) is used. We choose to use IndexedDB due to its good browser compatibility and flexible API, which helps to store information on local disk and its ability to query using different attributes. As name suggests, IndexedDB can index data using keys to provide fast retrieval of the information. Because the framework requirement is to provide two queries, by key and by author, two indexes are created. Both keys are not unique, so the API returns a list of results sorted by creation date or null in case of errors. Before saving the data, the controller checks if both key and ID are present. Note that, as mentioned previously, the ID is generated by DataController.

Background Page. When running in the Chrome app, the client has to manage its state in the background page. In this case the application is able to synchronize messages even when the program is not running in the foreground. This is an important part of the requirements, as clients might not always run the application. Initially we tried to construct the ‘DataController’ instance in

background page and then attempted to access it directly from the foreground application for querying and saving. Unfortunately, this approach did not work, since the foreground and background pages run in different contexts, and while it is possible to access primitive variables using ‘getBackgroundPage’ method, Chrome browser can only send JSON-serializable types between pages. Since we are using socket objects in ‘DataController’, it cannot be JSON-serialized properly; hence, it is not possible to have the ‘DataController’ in the foreground page. To overcome this challenge the ‘DataControllerClient’ and the ‘DataControllerReceiver’ were created. These two classes reside within different pages/contexts of the application. Because it’s impossible to directly interact with objects that are in the background page, JSON messages (that contain action information) are sent to the background page, which it receives and then executes the actions with DataController.

To pass information from the foreground page to the background page, we used ‘Chrome Message Passing’, where ‘DataControllerClient’ can send a JSON object with ‘action’ and ‘params’ attributes. The ‘action’ attribute points to the DataController method that should be called, whereas, the ‘params’ are parameters that need to be pass to the method. ‘DataControllerReceiver’ listens to the message in the background page, reads ‘action’ and executes method in same context with provided ‘params’.

Since the ‘DataControllerClient’ is for application use, that is why it is created on the front page. It has the very same API as ‘DataController’, with a difference that it delegates functionality to ‘DataController’ that resides in ‘DataControllerReceiver’. Figure 6 shows this behavior in case ‘DataControllerClient’ is instantiated in Chrome Application. If the ‘DataControllerClient’ is created in regular web applications, then there is no need for a background page; this is why the client creates a ‘DataController’ inside class and executes actions locally. For this reason, application can use ‘DataControllerClient’ and run the same code as a Chrome Extension or a Web application, and the ‘DataControllerClient’ can handle both cases without changing the code.

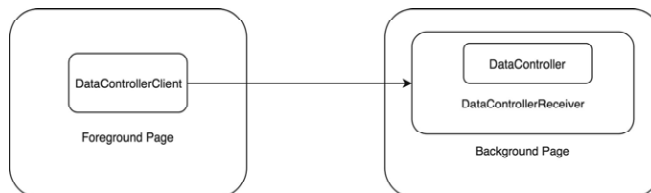


Fig. 6. Illustration of ‘DataController’ in Chrome Extension

6 Conclusion

To summarize, in this paper we set out to design and develop a P2P data-sharing framework that enables developers to create applications with powerful

data ownership and privacy features. The proposed framework is motivated by the rising issue of data ownership and privacy in the World Wide Web. To this end, we first provided a brief description of the state-of-art in data ownership and privacy. Drawing from the insights and solutions provided by these artifacts, we identified several requirements that when implemented as a framework could support users by providing them more data ownership and by reducing the role of servers in data exchange over the Web. The proposed framework is a simple tool that can be integrated into various kinds of applications. Our trials show that such a framework is undoubtedly useful for chat applications, where real-time data sending via peer-to-peer networks is critical. The framework is explicitly designed to secure user data and hence makes tampering with data extremely difficult. Applications based on the framework can create public channels where any user can query using application-defined keys and can share data both within closed groups and in public. Our implementation demonstrates data synchronization even when peers are offline. Additionally, the framework is built with multiple parts and services; and hence can be easily extended with custom functionalities. Developers are allowed to make changes to multiple parts of the framework and can, therefore, extend the functionalities of the system according to their requirements. Independent developers are also allowed to implement custom policies for data handling; whereas organizational implementations of the framework allows enforcement of user authentication on data sharing services, such as ‘Message Box’ and ‘Live-Rooms’, so that only people with specific access can use a service. As part of our future work, we plan to improve the framework further, based on support from the community. We would like to better understand developers’ needs for data management and evolve the proposed framework into a more useful tool, for both developers and end-users.

References

1. Bakir, V., McStay, A.: Fake news and the economy of emotions. *Digit. J.* **6**(2), 154–175 (2018). <https://doi.org/10.1080/21670811.2017.1345645>
2. Cadwalladr, C., Graham-Harrison, E.: Revealed: 50 million Facebook profiles harvested for Cambridge analytica in major data breach, March 2018. <https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election>. The Guardian. Accessed 13 Aug 2019
3. Capadisli, S., Guy, A., Verborgh, R., Lange, C., Auer, S., Berners-Lee, T.: Decentralised authoring, annotations and notifications for a read-write web with dokieli. In: Cabot, J., De Virgilio, R., Torlone, R. (eds.) *ICWE 2017*. LNCS, vol. 10360, pp. 469–481. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-60131-1_33
4. Dodson, B., Vo, I., Purtell, T., Cannon, A., Lam, M.: Musubi: disintermediated interactive social feeds for mobile devices. In: *Proceedings of the 21st International Conference on World Wide Web, WWW 2012*, pp. 211–220. ACM, New York (2012). <https://doi.org/10.1145/2187836.2187866>
5. Draheim, D., Felderer, M., Pekar, V.: Weaving social software features into enterprise resource planning systems. In: Piazzolo, F., Felderer, M. (eds.) *Novel Methods and Technologies for Enterprise Information Systems*. LNISO, vol. 8, pp. 223–237. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07055-1_18

6. Ford, B., Srisuresh, P., Kegel, D.: Peer-to-peer communication across network address translators. In: Proceedings of the Annual Conference on USENIX Annual Technical Conference, ATEC 2005, pp. 13–13. USENIX Association, Berkeley (2005). <http://dl.acm.org/citation.cfm?id=1247360.1247373>
7. Heitmann, B., Kim, J.G., Passant, A., Hayes, C., Kim, H.G.: An architecture for privacy-enabled user profile portability on the web of data. In: Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems, HetRec 2010, pp. 16–23. ACM, New York (2010). <https://doi.org/10.1145/1869446.1869449>
8. Isaak, J., Hanna, M.J.: User data privacy: Facebook, Cambridge analytica, and privacy protection. *Computer* **51**(8), 56–59 (2018). <https://doi.org/10.1109/MC.2018.3191268>
9. Kaplan, A.M., Haenlein, M.: Users of the world, unite! The challenges and opportunities of social media. *Bus. Horiz.* **53**(1), 59–68 (2010). <https://doi.org/10.1016/j.bushor.2009.09.003>
10. Kim, H.C.: Acceptability engineering: the study of user acceptance of innovative technologies. *J. Appl. Res. Technol.* **13**(2), 230–237 (2015). <https://doi.org/10.1016/j.jart.2015.06.001>
11. Knight, R.: Convincing skeptical employees to adopt new technology, August 2015. <https://hbr.org/2015/03/convincing-skeptical-employees-to-adopt-new-technology>. Harvard Business Review. Accessed 13 Aug 2019
12. Lazer, D.M.J., et al.: The science of fake news. *Science* **359**(6380), 1094–1096 (2018). <https://doi.org/10.1126/science.aao2998>
13. Mansour, E., et al.: A demonstration of the solid platform for social web applications. In: Proceedings of the 25th International Conference Companion on World Wide Web, WWW 2016 Companion, pp. 223–226. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva (2016). <https://doi.org/10.1145/2872518.2890529>
14. Pattanaik, V., Norta, A., Felderer, M., Draheim, D.: Systematic support for full knowledge management lifecycle by advanced semantic annotation across information system boundaries. In: Mendling, J., Mouratidis, H. (eds.) CAISE 2018. LNBP, vol. 317, pp. 66–73. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-92901-9_7
15. Richards, R.: Representational state transfer (REST), pp. 633–672. Apress, Berkeley (2006). https://doi.org/10.1007/978-1-4302-0139-7_17
16. Sambra, A., Guy, A., Capadislis, S., Greco, N.: Building decentralized applications for the social web. In: Proceedings of the 25th International Conference Companion on World Wide Web, WWW 2016 Companion, pp. 1033–1034. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva (2016). <https://doi.org/10.1145/2872518.2891060>
17. Sambra, A., Hawke, S., Berners-Lee, T., Kagal, L., Aboulmaga, A.: CIMBA: client-integrated microblogging architecture. In: Proceedings of the 2014 International Conference on Posters & Demonstrations Track, ISWC-PD 2014, vol. 1272, pp. 57–60. CEUR-WS.org, Aachen (2014). <http://dl.acm.org/citation.cfm?id=2878453.2878468>
18. Sambra, A.V., et al.: Solid: a platform for decentralized social applications based on linked data. Technical report, MIT CSAIL & Qatar Computing Research Institute (2016). <https://www.semanticscholar.org/paper/Solid-%3A-A-Platform-for-Decentralized-Social-Based-Sambra-Mansour/5ac93548fd0628f7ff8ff65b5878d04c79c513c4>

19. Story, H., Harbulot, B., Jacobi, I., Jones, M.: FOAF+SSL: RESTful authentication for the social web. In: CEUR Workshop Proceedings (2009)
20. Van Kleek, M., et al.: Social personal data stores: the nuclei of decentralised social machines. In: Proceedings of the 24th International Conference on World Wide Web, WWW 2015 Companion, pp. 1155–1160. ACM, New York (2015). <https://doi.org/10.1145/2740908.2743975>

Appendix 4

IV

V. Pattanaik, I. Sharvadze, and D. Draheim. A peer-to-peer data sharing framework for web browsers. *SN Computer Science*, 1(4), June 2020



A Peer-to-Peer Data Sharing Framework for Web Browsers

Analysis and Evaluation

Vishwajeet Pattanaik¹ · Ioane Sharvadze² · Dirk Draheim¹

Received: 20 April 2020 / Accepted: 18 June 2020 / Published online: 27 June 2020
© Springer Nature Singapore Pte Ltd 2020

Abstract

Concerns over data ownership and misuse of personal data over the Web have become increasingly widespread in recent years; especially, as most web service providers are moving towards closed silo-based platforms, making the web more and more centralized. This is concerning, because, as service providers move towards centralized data storage and management, end-users become more susceptible to loss of data ownership and misuse of personal data. While in recent years, quite a few solutions have been proposed to solve these issues, the issues themselves still prevail, primarily due to lack of acceptance. That said, in this paper, we build on our previously proposed browser-based Peer-to-Peer Data Sharing Framework. We first explain the requirements and design choices which we had to keep in mind while designing the framework. And then, we provide insights into how we evaluated the functionalities and security features of the framework, through lab experiments. Finally, we elucidate the direction in which we would like to develop the framework in the near future.

Keywords Data ownership · Decentralization · Human–computer interaction · Peer-to-peer · Social Web · Security · Web apps · WebRTC

Introduction

Since its inception nearly three decades ago, the Web has slowly but steadily become rather centralized; more so much especially in recent years [14, 20, 24]. Although, centralization is not necessarily bad; it, however, contradicts with the original goal behind the World Wide Web's (WWW) inception. The WWW was originally designed to be a decentralized network, viz., 'a common information space' for end-users to communicate with each other by sharing

information.¹ It was meant to be an open space where end-users could share any piece of data and information which they wanted to, thereby fostering creativity and innovation [25]. Unfortunately, with the rise in interest in social media platforms like Facebook, Instagram, Twitter, and YouTube [10] and the ever-rising number of web users worldwide; most popular social media platforms have opted to close their platforms into centralized data silos [20, 23]. Although, this centralization enables platform owners to provide polished and tailored experiences to their end-users; it also, however, opens up the end-users' personal data to the platform owners and stakeholders.

Due to the closed nature of said social media platforms, end-users are unknowingly forced into creating dedicated accounts on different platforms, thereby compelling the end-users to trust such providers to store and manage the users' personal data, and to rely on the platforms' services and resources for a more personalized user experience [20]. This, however, as recent literature suggests, has led to incidents where user data have been harvested [9] to improve machine learning algorithms, to manipulate user behaviour [1], and to spread misinformation and propaganda [2, 13, 25]. Such

This article is part of the topical collection "Future Data and Security Engineering 2019" guest edited by Tran Khanh Dang.

✉ Vishwajeet Pattanaik
vishwajeet.pattanaik@taltech.ee

✉ Ioane Sharvadze
ioane.sharvadze@gmail.com

Dirk Draheim
dirk.draheim@taltech.ee

¹ Information Systems Group, Tallinn University of Technology, Tallinn, Estonia

² Microsoft, Tallinn, Estonia

¹ W3C | The World Wide Web: A very short personal history.

incidents have prompted serious concerns, about data ownership, misuse of user data, and lack of secure interoperability between platforms owned by different organizations. And while several remarkable contributions [3, 4, 19–21] have been made to tackle these challenges, most of the said contributions are still underutilized.

A key reason for this lack of acceptance of such technologies is possibly the amount technical expertise and the initial implementation costs (with respect to time and money) required to migrate from a completely closed platform to a truly open one. To understand why end-users might or might not decide to adopt a technology would require further investigation; as Acceptability Engineering is a rather new field of research [11] which is still being developed. Drawing from Knight's work [12], however, we would like to argue that typical Internet users may choose not to adopt a technology simply because of lack of familiarity. And we are convinced that if end-users are provided with a secure means of sharing information: while enabling them to store their personal data on their own machines when using web-based platforms, and without the inconvenience of configuring or setting-up applications; most end-users, in general, might find it easier to share data online, while maintaining ownership of their data.

To this end, we proposed a novel framework [16] that aims to reduce the dependency on centralized servers and attempts to empower end-users with true data ownership. The proposed framework is designed as a browser-based platform-independent framework, which allows for data sharing between applications over peer-to-peer (P2P) networks. The framework provides end-users with complete data ownership, by allowing users to store their personal data on their personal machines and share it directly with other users, whenever required; thereby reducing the risk of data theft and monitoring. The proposed framework adds to our ongoing research [5, 15, 17] and aims to contribute to recent initiatives for 're-decentralizing the Web'. The framework attempts to eliminate the need for servers, as a means for data storage and sharing, and allows users to communicate with each other without the need for middle-ware. We designed the P2P framework as a 'generic system', increasing the scope of the framework. And finally, given the surge of interest in web-based applications and browser-based apps, we decided to implement the framework into a browser extension that provides Application Programming Interface (API)-based server-free communication between applications, thereby empowering not only end-users but also developers.

Now, building on our previous work [16], in this paper, we delve deeper into the analysis of the design choices, security aspects, and evaluation of the said framework. We start by briefly discussing the state-of-the-art that inspired and influenced the framework; and then touch on the challenges, we would like to tackle by means of the framework. We

re-introduce the functional requirements that we want the framework to fulfill and reaffirm the importance and limitations of connecting peers behind NATs. In "P2P data sharing framework", we provide an overview of the proposed framework; following which, in "Implementation and security features", we elucidate the added layers of security that we integrated into the model to ensure the confidentiality of user's personal data. In "Lab experiments", we discuss the lab experiments which we conducted, to test the functional requirements, P2P capacity, and security layers of the proposed framework. Finally, we talk about the next steps which we would like to take to further extend and improve the usability of the framework; and conclude by summarizing our overall findings.

Background and Related Work

As mentioned in "Introduction", in recent years, numerous scientific artifacts have been proposed to tackle data ownership and privacy issues. Among said artifacts, our work has been primarily influenced by platforms including Musubi [4], CIMBA [20], Solid [14, 21], and Dokieli [3]. Each of these platforms was designed to allow users to create and share data via secure networks.

The Musubi [4] platform was proposed by Dodson et al. in 2012 and is designed as a mobile social application platform that allows users to share real-time feeds over mobile devices. The platform allows users to interact with their friends directly through their address books, while ensuring data security and privacy with the help of end-to-end public-key encryption. Also, the platform enables data ownership, by storing user data on their personal mobile devices. Unfortunately, since the platform is primarily designed for mobile devices, it does not support direct P2P connections, as it is not possible to establish a true P2P network over a 3G network. Musubi, therefore, relies on a centralized service Trusted Group Communication Protocol [4].

CIMBA or Client-Integrated Micro-Blogging Architecture [20] was proposed by Sambra et al. in 2014. The platform is designed as a decentralized social web platform that attempted to decouple an application's web server from the user's database. The architecture allows users to choose where they would like to store their personal data, and uses WebID² [8, 22] and WebID-TLS³ to identify users and to authenticate requests. This enables user data ownership, as users can decide what applications can access what part of their data.

² W3C | WebIDs and the WebID Protocol.

³ W3C | WebID Authentication over TLS (editor's draft).

The Dokieli [3] platform was proposed by Capadisi et al. in 2017. The platform is a decentralized browser-based authoring and annotation tool, which allows users to retain the ownership of their data while allowing for social interactions. Finally, the SOLID, i.e., the Social Linked Data platform [14, 21], was proposed by Samba et al. in 2016. SOLID is a decentralized platform for social web applications and builds on the findings of CIMBA. Unlike conventional web applications where users are forced to store personal data of servers controlled by web application providers, SOLID users are required to store their data in personal pods (i.e., online data stores). Similar to CIMBA, SOLID users are identified using WebIDs [8, 22] and have complete control over how their data is accessed. Finally, the platform uses Resource Description Framework (RDF)-based resources to exchange data between applications and pods [14, 21].

Although the above-mentioned platforms support data ownership, however, they still rely on non-user devices to store users' personal data. Also, since these solutions require users to carry out technical activities, such as finding/configuring hosting services and setting up pods; a typical non-technical user can find these tasks overwhelming, thereby preventing them from adopting these technologies. Hence, we are of the opinion that one could improve the users' overall experience by removing the need for any configuration and by storing the users' data onto their own personal computers.

Challenges

Drawing from these issues, and motivated by the lack of a generic, application-independent data sharing framework for P2P networks, we set out to tackle the following key challenges [16]:

- *Enabling P2P network establishment between web browsers* We investigated how P2P networks can be established over web browsers, and examined how [Socket.IO](#) and [WebRTC](#) could be used to establish communication in P2P networks. Additionally, we delved into the question of how we can solve the NAT traversal [6] problem using JavaScript.
- *Creating a server-less P2P network* As the literature suggests, most decentralized web solutions still rely on servers to at least to some extent. We explored whether P2P networks can be established in a truly server-less environment. If not, we set out to provide alternative approaches to reduce dependency on servers.
- *Enabling message exchange when peers are offline* Since the proposed framework needed to be deployed over P2P networks, we investigated how we could enable message exchange with peers that are offline, at the time of mes-

sage sending. Also, we devised a fallback mechanism that could still deliver messages in such a scenario.

Functional Requirements

In addition to the above-mentioned challenges, the proposed framework is designed around the Tippiance platform [15, 17] (see “[Introduction](#)”); and therefore, the framework is needed to fulfill a specific set of functional requirements. It is important to understand that most of the said requirements would have to be fulfilled for the framework to be useable for other social web applications, as well. Primarily, the framework should support data sharing and control activities (i.e., maintaining privacy and security), and should be simple enough to hide data management complexities from application developers, making it easy to understand and implement.

In general, the said framework should fulfill the following functional requirements:

- *Public and private data sharing* The framework should allow users to share data not only one-to-one (i.e., privately, where only the sender and receiver have access to the data), but also one-to-many (i.e., publicly, where the data are accessible to every user on the platform).
- *Private data sharing in groups* This requirement is rather a combination of the ‘Public & Private Data Sharing’ requirement, where the framework should allow users to share data one-to-many; however, the ‘many’ should belong a specified community or group or circle. And hence, platform users who are not a part of the said group should not have access to the data.
- *Caching data for offline peer* This is a crucial requirement, wherein the P2P framework should allow for data delivery to peers, who are offline at the instant when the data are shared. Therefore, the framework needs to have a mechanism in place, to securely hold the data until the receiver comes back online.
- *Storing data locally* The framework should allow local storage, so users can access the sent or received data even when they are offline (or, not connected to the network), or in case the shared data are lost or deleted by the sender.
- *Security and integrity* A second key requirement of the framework is that it should be based on the standard principles of information security. That is, malicious users should not be allowed to tamper with the privately shared/stored data.
- *Technical requirements* Finally, the framework should be designed in a way that it could be encoded into a browser-based web applications or browser extensions. And, since web applications and extensions may present

different technical requirements for the framework, the framework should be adaptable to both.

Limitations with Peers Behind NATs

Based on Ford et al.'s work [6], we know that it is not possible to establish a P2P communication between users without a third-party server. This is because most devices on the Web are hidden behind Network Address Translators (NAT), which means that most web users do not have a unique Internet Protocol (IP) address. Instead, they have a unique local address within their network, which is provided to them by the NATs. The NATs are responsible for translating users' local IP into unique public IP and port configurations, when communicating with systems outside the local network. This implies that multiple users behind a single NAT would virtually have the same public IP, but with different ports configurations. And if a device gets disconnected from the network, it might end up receiving a completely different IP and port configuration, thus forbidding incoming connection requests to the device. This issue, however, can be overcome if peers request connection to one another at the same time. In such a scenario, Hole Punching would occur and the NAT could most likely (64% cases for TCP connections [6]) enable a Peer-to-Peer connection. Unfortunately, the remaining 36% cases could lead to situations, where it might not be possible to share data between peers without a middleware server. This is a key limitation that we would like to overcome with our framework.

P2P Data Sharing Framework

Building on the requirements and limitations described in the previous sections, in this section, we briefly discuss the components of our P2P data sharing framework.

Using Servers

As discussed in "Limitations with peers behind NATs", to establish a P2P connection, peers must have access to each others' IP addresses. If the IP addresses are known, then both peers must request for connection simultaneously. Since the scenario cannot be guaranteed, especially over mobile and public WiFi connections, it is imperative to have a relay server. Such a server allows peers to share their public IP addresses and thus assists in establishing P2P connections.

The framework [16] works around this challenge with the help of Google's Web Real-Time Communications protocol (WebRTC). WebRTC is a set of APIs that is implemented by most web browsers and, therefore, fulfills the browser supporting requirements of the framework. Since WebRTC is only an API, we developed a signaling implementation

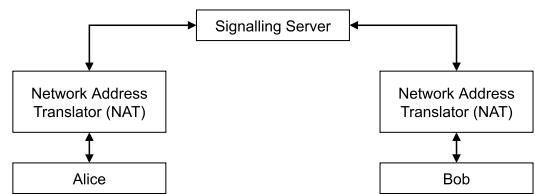


Fig. 1 Illustration of typical signalling behind NAT

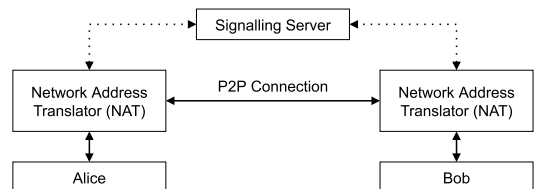


Fig. 2 Illustration of P2P connection establishment behind NAT

(into the framework) that is used to establish WebRTC connections. We used Socket.IO [11] for this, as the library is well documented and widely popular. In our current implementation, peers are required to first connect to the signaling server; and once two peers have exchanged their IP information, they can start sharing data with each other directly *without the server* (see Figs. 1, 2).

Apart from establishing P2P connections, the framework uses a relay (i.e., signaling) server in two more situations. First, for public data holding, i.e., when data that are shared in public is stored on a public server. Since all members of the network might not be familiar with one another, if a user decides to share some data with everyone on the network, the public server stores a copy of the data. Peers who might be interested in the data could then access the publicly shared data as per their convenience.

Second, for sharing data with offline peer. Like we discussed earlier, in case a peer decides to share data with an offline peer, the receiver would have to come back online to get the message. To make sure that the receiver gets the message, the storage server temporarily holds a copy of the shared data. Once the receiver reconnects to the relay server, the data are transferred to the intended receiver and then removed from the storage server.

To reiterate, the server in the framework is imperative but only as a fallback mechanism.

Server Architecture

Taking into account the challenges which we described in "Challenges", to ensure privacy and data ownership to

Fig. 3 Illustration of the message sending process in the proposed framework

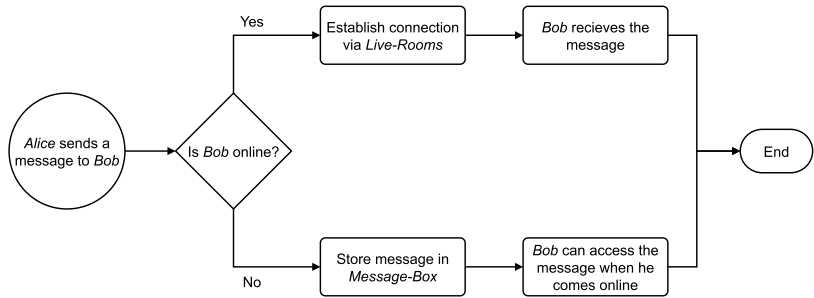
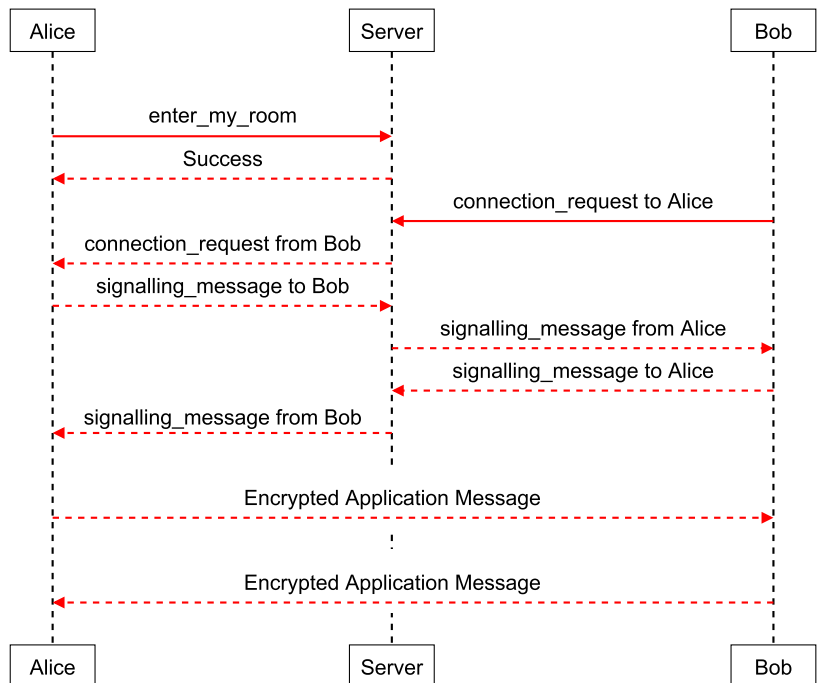


Fig. 4 Illustration of the P2P network establishment process in the proposed framework



end-users, the server within the framework is not designed to be application-specific, its APIs are implemented separately, and its role in the message exchange process is kept to a bare minimum. Finally, the said server is designed, so that it is application-independent and has negligible access to the hosted data.

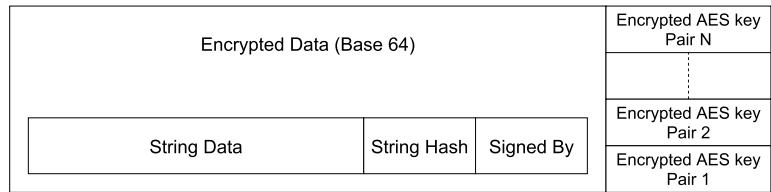
In our implementation, we designed two different services for said server. The first service allows peers to establish a connection via Live-Rooms, while the second service supports P2P message exchange (with offline peers) via Message-Box (as illustrated in Fig. 3).

The Live-Rooms service enables quick (real-time) data sharing between peers, and is designed as a common space where users gather and wait for incoming connections,

whereas the Message-Box service plays a more important role within the framework. It fulfills six key functionalities: (1) lists message-ID's for user; (2) downloads messages by message-IDs; (3) stores shared messages; (4) lists public messages by keys; (5) downloads public messages by keys; and (6) saves public messages with keys. Among these functionalities, the first three are required for message relay, while the remaining are required for storing/querying public data shared by the users. In the current implementation, we use the REST [18] client-server architecture, for the same.

Finally, the framework also includes: a Client-Side interface for application developers, WebRTC configurations for assisting in establishing a P2P connections (see Fig. 4), and

Fig. 5 Structure of encrypted data in the proposed framework



APIs to exchange messages between the Live-Rooms and Message-Box services [16].

Implementation and Security Features

The Live-Rooms, Message-Box, and Client Library components of the framework are designed and implemented as distinct entities. Doing so, it allowed us to extend the functionalities of the components without modifying other modules of the framework. The implementation of all of these modules is available as a GitHub repository.

In the current implementation, the Live-Rooms service is implemented using the Socket.IO library, while the Message-Box service is implemented using the Express.js JavaScript framework. For the Data Controller, we opted for Indexed DB, due to its stable browser compatibility and flexible API. A more detailed view of the framework’s implementation can be found in our previous work [16].

The Client Library component of the implementation is responsible for sending/receiving data over P2P network and for storing shared data in Message-Box. Since the proposed framework is designed to be generic, the Client Library could be treated as service maintained by third parties. This means that data being handled by the service could be susceptible to tampering or theft. Keeping this in mind, we implemented end-to-end RSA [26] encryption to our artifact. The user’s unique id is thus treated as a public key, making the data accessible (meaningfully) only to its user. Unfortunately, since public key encryption can restrict the size of shared data, we decided to first encrypt user data with AES–CBC encryption (that allows for an unlimited size of data); after which the AES–CBC key pair is encrypted using the data owners’ unique key. The AES–CBC encryption process in the system is done with randomized initial vectors and keys. This is vital, as this prevents an attacker from deducing if multiple messages have the same content. Doing so allows the system to not only secure unlimited size of data, but also makes sure that only the owner of the data can access the complete data. Finally, as an added layer of protection, since every exchanged message uses public-key signatures; peers receiving the message can verify and ensure if the message has been sent by a known peer or an unknown attacker.

Since we implemented the framework as a Google Chrome browser extension, we choose to use the Forge JavaScript library. The library has a fully native implementation of Transport Layer Security (TLS) and provides a common set of tools for encryption/decryption, key generation, and signatures. Before integrating security into the application, we defined the Security.js, which helps to create abstract APIs for the application, making it easier to use in different parts of the application.

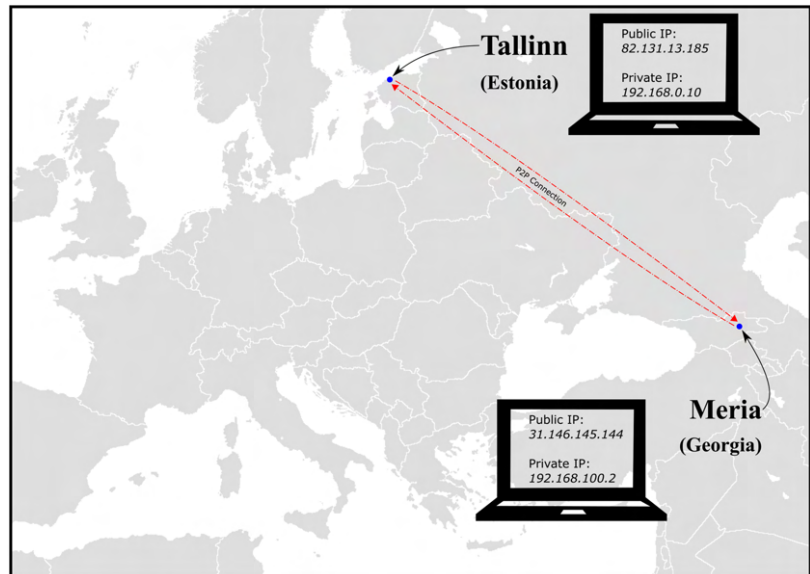
Encryption–Decryption Process

A simplified explanation of how the message exchange process is secured is as follows. When a sender decides to send a message to the receiver, the system starts by signing the SHA-1 hash of the data (this helps to reduce the data size). The signed data are then attached to the sender’s public key; and the object is converted into a string and then encrypted. The Forge library encrypts the string and outputs binary data. Unfortunately, it is inconvenient to handle binary data in Js and the same cannot be sent over HTTP (which is a text protocol). Therefore, the encrypted data are then converted into Base64 encoding and the stringified AES key is encrypted for the sender’s and receivers’ access key.

To read the message, on the receiver side, the system first finds the receiver’s AES encrypted key, and then decrypts the AES key with the users’ private key. At this point, the parsed JavaScript object is converted from Base64 string into binary, then decrypted with AES-CBC algorithm, and then validated to verify the identity of the sender.

Although it is common knowledge that public-key encryption is secure (at least as long as an attacker only has access to the public key), Hardesty [7] found that public-key encryption schemes can be weakened by Chosen-Ciphertext Attacks (CCAs); where an attacker has samples of successful decryptions. Hence, adding the extra layers of encryption/decryption mentioned above (also see Fig. 5) helps to ensure that the messages being passed between peers, or being stored on the relay server have a lesser chance of being decrypted by unauthorized users.

Fig. 6 Illustration of P2P network testing during lab experiments



Lab Experiments

To validate if the proposed framework [16] performs as we intended, and to verify if the expected functional requirements have been fulfilled, we encoded the framework into an artifact. We deployed a web application on the Heroku Cloud Application Platform; and the database for the application was established in mLab database platform, which was then connected to the Message-Box service and the Heroku server. The demo application itself was designed as a simple chat application, which is accessible both as a Google Chrome browser extension and as a website.

For lab experiments, we carried out functional system testing of the deployed artifact. The intention of the experiments was to explicitly check whether the designed artifact fulfilled the functional requirements of the framework (as mentioned in Sect. 2.2). The experiments were short and were only carried out as a sanity check. To conduct the said experiments, we used the deployed web application on five different clients over different local networks. The five clients/users sent messages of varying lengths (trying to replicate short conversations), and verified if the application was able to send/receive messages in different scenarios; for example: when clients were online, or when the clients (receivers) were offline. To check the group sharing features, we also clubbed the users in two groups of twos and threes.

During our lab experiments, we found that the application performed as expected. Users were able to exchange messages successfully via Live-Rooms when both peers were online. And in situations where peers (receivers) were not

online, the messages were stored in the Message-Box. When the receiving peers reconnected to the network, the messages were successfully moved from the server to the users' local storage. For reproducibility, the demo Chrome extension is also available as a GitHub repository.

We would like to reiterate that our goal was only to validate whether the framework fulfilled the functional requirements or not. Therefore, we choose to do the experiments only in lab settings, as this allowed us to focus on current goals. In future, we plan to do more exhaustive studies on the response times and time-space complexities of the encryption/decryption process; however, we understand that these factors closely rely on the computational capabilities of users' personal computers (i.e., client PCs), the Internet speed (at the clients' side), and the computational capabilities of the relay server; all of which are external factors, that the proposed framework currently does not tackle.

Testing P2P Network Establishment

In a separate set of lab experiments, we verified the P2P data transfer capability of the framework by sending messages across two different clients; both behind different NAT servers. For these experiments, one of the peers was connected to the relay server from Tallinn, Estonia, while the second client was connected from Meria, Georgia (as illustrated in Fig. 6). We verified that both peers were behind NATs by checking the systems' public and private IP addresses.

As shown in Fig. 6, both peers were situated in different countries, and hence, it meant that the Internet Service

Fig. 7 Screenshot of a message stored in the *Message-Box*, as viewed in mLab database dashboard



Providers (ISP) could have multi-layer NAT networks. We found that both peers were successful in establishing a P2P connection even behind their NATs, thereby validating our claims on P2P network establishment feature of the framework.

Testing Message Security

Since testing security of the system can be a really challenging task in the real-life environment, we decided to conduct only simple lab experiments to check if invalid/tempered messages could still be delivered via our artifact. We choose to test the following abstract scenario: let us say, Alice leaves a message for a Bob, while Bob is offline. In such a scenario, the ‘Message-Box’ server should store the message temporarily, until Bob is back online and accepts the message. Now, when Bob returns, he should only receive the message if the state of the signed message is the same as the one sent by Alice.

We tested the said scenario on multiple occasions, by changing the sent message using mLab database dashboard. And each time, the modified message was rejected on the receiver’s side, i.e., the receiver never got a message that was modified after the sender had signed and sent it. Figure 7 shows a screenshot of one such message as viewed in mLab database dashboard. Based on these brief experiments, we concluded that the multilayered security of the proposed framework successfully accomplished its goals of securing messages being stored on the relay server.

Conclusion

The criticality of moving towards a truly re-decentralized Web cannot be understated; especially, given the rising concerns over data ownership and privacy. In this paper, we set out to analyze and evaluate a peer-to-peer framework for data sharing over web browsers. Our goal in this paper was to extend on our previous contribution of providing an easy to implement solution for web developers, which could empower a web application’s end-user with complete data ownership.

We briefly described the key shortcomings in the current state-of-art, and then discussed our previously proposed novel framework that enables user data ownership by reducing the role of servers in data exchange, to a bare minimum. We investigated the functionality of the framework by means of a ‘toy’ chat application, while retaining the framework’s genericness. We then delved into the security features, and discussed how the framework was designed keeping the user’s privacy and security in mind. And, through lab experiments, we showed that the proposed framework successfully fulfilled the requirements and tackled the challenges that we set out to accomplish.

Reflecting on the discussed experiments, we understand that we need to explore how the framework would perform under CCAs and other targeted attacks; however, we believe that in its current preliminary state, the proposed framework provides apt layers of security already (i.e., by means of its P2P capability, and its multiple layers of encryption/decryption). That said, as part of our future work, we plan to develop the proposed framework into a

more useful tool, for both developers and end-users, by integrating it into an online platform for the crowd. In the next iteration of the framework, we would like to dig deeper into the current security shortcomings of the framework; and would like to investigate the reliability of the multiple security layers from a cybersecurity perspective. Finally, to conclude, we would like to reiterate that the proposed P2P framework is only but a small step towards the vision of an open and decentralized web; where all web users could collaborate and innovate, without worrying about issues such as monitoring, privacy, and data theft.

Compliance with Ethical Standards

Conflicts of interest The authors declare that they have no conflict of interest.

References

- Bakir V, McStay A. Fake news and the economy of emotions. *Digit J*. 2018;6(2):154–75. <https://doi.org/10.1080/21670811.2017.1345645>.
- Cadwalladr C, Graham-Harrison E. Revealed: 50 million facebook profiles harvested for cambridge analytica in major data breach. <https://www.theguardian.com/news/2018/mar/17/cambidge-analytica-facebook-influence-us-election> (2018). *The Guardian* (online); Accessed 13 Aug 2019.
- Capadisli S, Guy A, Verborgh R, Lange C, Auer S, Berners-Lee T. Decentralised authoring, annotations and notifications for a read-write web with dokiel. In: Cabot J, De Virgilio R, Tolorne R, editors. *Web engineering*. Cham: Springer International Publishing; 2017. p. 469–81.
- Dodson B, Vo I, Purtell T, Cannon A, Lam M. Musubi: dis-intermediated interactive social feeds for mobile devices. In: *Proceedings of the 21st international conference on world wide web, WWW '12*. Association for Computing Machinery, New York, NY, USA; 2012. p. 211–220. <https://doi.org/10.1145/2187836.2187866>.
- Draheim D, Felderer M, Pekar V. Weaving social software features into enterprise resource planning systems. In: Piazzolo F, Felderer M, editors. *Novel methods and technologies for enterprise information systems*. Cham: Springer International Publishing; 2014. p. 223–37.
- Ford B, Srisuresh P, Kegel D. Peer-to-peer communication across network address translators. In: *Proceedings of the annual conference on USENIX annual technical conference, ATEC '05*. USENIX Association, Berkeley, CA, USA; 2005. p. 13. <http://dl.acm.org/citation.cfm?id=1247360.1247373>
- Hardesty L. Beefing up public-key encryption. *MIT News*; 2013. <https://news.mit.edu/2013/beefing-up-public-key-encryption-0215>. Accessed 27 May 2020.
- Heitmann B, Kim JG, Passant A, Hayes C, Kim HG. An architecture for privacy-enabled user profile portability on the web of data. In: *Proceedings of the 1st international workshop on information heterogeneity and fusion in recommender systems, HetRec '10*. Association for Computing Machinery, New York, NY, USA; 2010. p. 16–23. <https://doi.org/10.1145/1869446.1869449>.
- Isaak J, Hanna MJ. User data privacy: Facebook, cambridge analytica, and privacy protection. *Computer*. 2018;51(8):56–9. <https://doi.org/10.1109/MC.2018.3191268>.
- Kaplan AM, Haenlein M. Users of the world, unite! the challenges and opportunities of social media. *Bus Horiz*. 2010;53(1):59–68. <https://doi.org/10.1016/j.bushor.2009.09.003>.
- Kim HC. Acceptability engineering: the study of user acceptance of innovative technologies. *J Appl Res Technol*. 2015;13(2):230–7. <https://doi.org/10.1016/j.jart.2015.06.001>.
- Knight R. Convincing skeptical employees to adopt new technology. <https://hbr.org/2015/03/convincing-skeptical-employees-to-adopt-new-technology> (2015). *Harvard Business Review* (online); Accessed 13 Aug 2019
- Lazer DMJ, Baum MA, Benkler Y, Berinsky AJ, Greenhill KM, Menczer F, Metzger MJ, Nyhan B, Pennycook G, Rothschild D, Schudson M, Sloman SA, Sunstein CR, Thorson EA, Watts DJ, Zittrain JL. The science of fake news. *Science*. 2018;359(6380):1094–6. <https://doi.org/10.1126/science.aao2998>. <https://science.sciencemag.org/content/359/6380/1094>.
- Mansour E, Sambra AV, Hawke S, Zereba M, Capadisli S, Ghanem A, Aboulnaga A, Berners-Lee T. A demonstration of the solid platform for social web applications. In: *Proceedings of the 25th international conference companion on world wide web, WWW '16 Companion*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE; 2016. p. 223–226. <https://doi.org/10.1145/2872518.2890529>.
- Pattanaik V, Norta A, Felderer M, Draheim D. Systematic support for full knowledge management lifecycle by advanced semantic annotation across information system boundaries. In: Mendling J, Mouratidis H, editors. *Information systems in the big data era*. Cham: Springer International Publishing; 2018. p. 66–73.
- Pattanaik V, Sharvadze I, Draheim D. Framework for peer-to-peer data sharing over web browsers. In: Dang TK, Küng J, Takizawa M, Bui SH, editors. *Future data and security engineering*. Cham: Springer International Publishing; 2019. p. 207–25.
- Pattanaik V, Suran S, Draheim D. Enabling social information exchange via dynamically robust annotations. In: *Proceedings of the 21st international conference on information integration and web-based applications & Services, iiWAS2019*. Association for Computing Machinery, New York, NY, USA; 2019. p. 176–184. <https://doi.org/10.1145/3366030.3366060>.
- Richards R. Representational state transfer (REST). Berkeley: Apress; 2006. p. 633–72. https://doi.org/10.1007/978-1-4302-0139-7_17.
- Sambra A, Guy A, Capadisli S, Greco N. Building decentralized applications for the social web. In: *Proceedings of the 25th International Conference Companion on World Wide Web, WWW '16 Companion*; 2016. p. 1033–1034. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE. <https://doi.org/10.1145/2872518.2891060>.
- Sambra A, Hawke S, Berners-Lee T, Kagal L, Aboulnaga A. Cimba: Client-integrated microblogging architecture. In: *Proceedings of the 2014 international conference on posters and demonstrations track, vol 1272, ISWC-PD'14, 2014*; p. 57–60. CEUR-WS.org, Aachen, DEU
- Sambra AV, Mansour E, Hawke S, Zereba M, Greco N, Ghanem A, Zagidulin D, Aboulnaga A, Berners-Lee T. Solid: A platform for decentralized social applications based on linked data. Tech. rep. MIT CSAIL & Qatar Computing Research Institute; 2016. http://emansour.com/research/iusail/solid_protocols.pdf. Accessed 27 May 2020.
- Story H, Harbulot B, Jacobi I, Jones M. FOAF+SSL: RESTful authentication for the social Web. In: *Proceedings of the first workshop on trust and privacy on the social and semantic Web (SPOT2009)*, CEUR workshop proceedings, Heraklion, Greece, June 2009, p. 1–12. <http://ceur-ws.org/Vol-447/paper5.pdf>.

23. Tomaiuolo M, Mordonini M, Poggi A. A p2p architecture for social networking. In: Applying integration techniques and methods in distributed systems and technologies. IGI Global. 2019; p. 220–245. <https://doi.org/10.4018/978-1-5225-8295-3.ch009>.
24. Van Kleek M, Smith DA, Murray-Rust D, Guy A, O'Hara K, Dragan L, Shadbolt NR. Social personal data stores: The nuclei of decentralised social machines. In: Proceedings of the 24th international conference on world wide web, WWW '15 Companion. Association for Computing Machinery, New York, NY, USA; 2015. p. 1155–1160. <https://doi.org/10.1145/2740908.2743975>.
25. Verborgh R. Re-decentralizing the Web, for good this time. In: Seneviratne O, Hendler J, editors. Linking the World's Information: Tim Berners-Lee's Invention of the World Wide Web. ACM (2020). <https://ruben.verborgh.org/articles/redecentralizing-the-web/>
26. Wardlaw WP. The rsa public key cryptosystem. In: Joyner D, editor. Coding theory and cryptography. Berlin: Springer; 2000. p. 101–23.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Appendix 5

V

S. Suran, V. Pattanaik, and D. Draheim. Frameworks for collective intelligence: A systematic literature review. *ACM Comput. Surv.*, 53(1), Feb. 2020

Frameworks for Collective Intelligence: A Systematic Literature Review

SHWETA SURAN, VISHWAJEET PATTANAİK, and DIRK DRAHEIM, Tallinn University of Technology, Estonia

Over the last few years, Collective Intelligence (CI) platforms have become a vital resource for learning, problem solving, decision-making, and predictions. This rising interest in the topic has led to the development of several models and frameworks available in published literature. Unfortunately, most of these models are built around domain-specific requirements, i.e., they are often based on the intuitions of their domain experts and developers. This has created a gap in our knowledge in the theoretical foundations of CI systems and models, in general. In this article, we attempt to fill this gap by conducting a systematic review of CI models and frameworks, identified from a collection of 9,418 scholarly articles published since 2000. Eventually, we contribute by aggregating the available knowledge from 12 CI models into one novel framework and present a generic model that describes CI systems irrespective of their domains. We add to the previously available CI models by providing a more granular view of how different components of CI systems interact. We evaluate the proposed model by examining it with respect to six popular, ongoing CI initiatives available on the Web.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Information systems** → **Crowdsourcing**; **Collaborative and social computing systems and tools**; • **Human-centered computing** → **Collaborative and social computing theory, concepts and paradigms**; **Collaborative and social computing systems and tools**; *Human computer interaction (HCI)*;

Additional Key Words and Phrases: Collective intelligence, crowdsourcing, human computer interaction, Web 2.0, wisdom of crowds, systematic literature review

ACM Reference format:

Shweta Suran, Vishwajeet Pattanaik, and Dirk Draheim. 2020. Frameworks for Collective Intelligence: A Systematic Literature Review. *ACM Comput. Surv.* 53, 1, Article 14 (February 2020), 36 pages.
<https://doi.org/10.1145/3368986>

1 INTRODUCTION

The concept of “Collective Intelligence” (CI) (i.e., collaborative problem solving and decision-making) has been a keen interest of researchers ever since the 18th century [41, 63]. Since this period, the different applications of CI and its associated concepts have extended throughout a wide spectrum of research domains ranging from sociology, psychology, biology, management, and economics to computer science, among many others [50]. In our work, we focus on CI in Information and Communications Technology (ICT), and therefore, we adhere to the widely accepted formal definition of CI in the ICT domain, proposed by Pierre Lévy in 1995 [43]. Lévy defined CI as

Authors’ addresses: S. Suran, V. Pattanaik, and D. Draheim, Tallinn University of Technology, Akadeemia tee 15a, Tallinn, 12616, Estonia; emails: {shweta, vishwajeet.pattanaik, dirk.draheim}@taltech.ee.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2020 Association for Computing Machinery.

0360-0300/2020/02-ART14 \$15.00

<https://doi.org/10.1145/3368986>

a “form of universally distributed intelligence, constantly enhanced, coordinated in real time, and resulting in the effective mobilization of skills” [43]. Some of the CI platforms of the early period include WikiWikiWeb, Experts-Exchange, and Google [50]. Since then, advancements in ICT technologies like Web 2.0 [65, 71], Semantic Web [28, 44], and Crowdsourcing [7, 17] have enabled and drastically eased large-scale collaborations over the Internet, leading to the development of well-known CI platforms like WaterWiki [16, 62], Climate CoLab [34, 51], DDtrac [26, 27], WikiCrimes [19, 68], and Goldcorp [4], which facilitate knowledge sharing, problem-solving, and decision-making among individual users and groups, through web-based interactions and collaborations.

The success of these systems can be credited to their underlying architectures or frameworks (hereinafter referred to as “models”). Unfortunately, most of these models are often defined using system-specific elements, principles, attributes, requirements, or their combinations [39], and are based on specific problems [21]. Since each of these CI systems is designed for a specific problem or use-case, the models proposed for these systems are often presented as completely different entities. However, comparing these models shows that although each new CI system and model expands on our current understanding of CI, nevertheless many of these systems bear a few similarities [48]. Sadly, this abundance of diverse knowledge has not yet led to the development of a unified CI model [13, 56, 67] that can support the development of new CI systems based on systematic knowledge rather than intuition [39]. Also, many of the existing CI systems are proprietary and are therefore not available in scientific literature. And, systems that are described in scientific literature, focus more towards the theoretical foundations, usability, and future applications of collective intelligence [21], rather than focusing on the implementation [39]. This lack of well-defined and systematic knowledge about the architecture and principles of the underlying CI systems has led to a reproducibility crisis.

In order to achieve comprehensive knowledge of CI systems, it is imperative that we extensively investigate published scientific literature irrespective of the so-called proposed models. We are convinced that although different CI systems are defined in different ways, they must share more than just a few common characteristics. And, identifying these characteristics could help us to achieve a unified formal model for designing CI systems, irrespective of their application. To this end, we contribute by conducting a first of its kind Systematic Literature Review (SLR) of CI models. In this SLR, we extensively investigate the characteristics of 12 CI models, selected from a pool of 219 scientific publications. And, based on the results of our review, we develop a novel framework that can be utilized to understand existing CI systems. The proposed framework provides a generic model and a set of requisites that would enable creation of novel CI systems, regardless of their domains. This is achieved by exhaustively combining all attributes of the studied CI models into the proposed framework.

Additionally, to better explain the functioning of CI systems with respect to the proposed framework, we examine the different components of six ongoing CI projects: CAPSELLA, hackAIR, openIDEO, Climate CoLab, WikiCrimes, and Threadless.

In particular, we aim at answering the following research questions:

RQ1: What are the underlying models of existing CI systems? What are the common terminologies used to describe CI models? What are their components? And, how are these components associated to each other?

RQ2: Do any of the available CI models appropriately define all CI systems, irrespective of their applications? Can these models be used to create CI systems for novel challenges?

RQ3: If not, then can we somehow combine the available knowledge of CI models and systems to create a unified model that could define all CI systems?

The article is structured as follows. In Section 2, we describe the research methodology used for conducting this SLR. Section 3 presents a brief summary of the selected studies, followed by the aggregated list of terminologies used to describe CI systems in Section 4. In Section 5, we present a novel framework for CI systems and evaluate our generic theoretical CI model by means of comparative case studies in Section 6. Finally, Section 7 presents the threats to validity of the SLR and Section 8 concludes by summarizing the key findings of this article and provides insights for future research.

2 RESEARCH METHODOLOGY

To answer the research questions mentioned in Section 1 through a transparent and objective approach, we decided to conduct this review based on Kitchenham’s “Guidelines for performing Systematic Literature Reviews in Software Engineering” [37]. A SLR summarizes, critically appraises, and identifies valid and applicable evidence in available research by using explicit methods to perform thorough literature search [9, 37, 66]. Based on Kitchenham’s guidelines, we perform this SLR in five stages:

- (1) Search Strategy
- (2) Study Selection
- (3) Study Quality Assessment
- (4) Data Extraction
- (5) Data Synthesis

2.1 Search Strategy

Based on the previously identified research questions, we selected a set of search terms. We then used the combination of these search terms to look for relevant research articles in different academic databases. After this, we applied the inclusion criteria on the identified articles and short-listed the most relevant articles (which we refer to as “Primary Studies”). Following Kitchenham’s guidelines, we then evaluated the primary studies using the quality assessment criteria. And finally, the selected studies were investigated in the data extraction and synthesis stages of the SLR.

2.1.1 Search Terms. As researchers often use the terms “Crowdsourcing” and “Wisdom of Crowds” as synonyms for CI [39, 64], we decided to use all three keywords as the primary search terms. And, for the secondary search terms, we used keywords such as model, framework, and others that are commonly used to describe ICT systems. In order to construct the search string we used the following guidelines provided by Kitchenham and Charters [37]:

- (1) Derive search terms from research questions and from initial literature review.
- (2) Identify synonyms for search terms from scientific literature.
- (3) Use the Boolean “AND” and “OR” to link search terms and their synonyms.

The list of identified primary and secondary search terms, and the resulting search string are presented in Table 1.

2.1.2 Academic Databases. The resulting search string was used to search for pertinent articles in four different databases, namely, *ACM Digital Library*, *IEEE Xplore*, *ScienceDirect (Elsevier)*, and *Springer*. The search was restricted to articles published since the year 2000; because, the first popular Web 2.0 based CI platform ‘Goldcorp’ and ‘Threadless’ were launched in the same year [18, 52, 83]. It was only after this period that CI systems became popular and were recognized as a significant area of research in ICT. In order to identify relevant books, technical reports, and theses, we also conducted a manual search on Google Scholar.

Table 1. Search Terms Identified Based on Research Questions

Primary Search Terms	collective intelligence, wisdom of crowds, crowdsourcing
Secondary Search Terms	model, framework, architecture, requirements, principles, attributes, properties
Search String	("collective intelligence" OR "wisdom of crowds" OR "crowdsourcing") AND ("model" OR "framework" OR "architecture" OR "requirements" OR "principles" OR "attributes" OR "properties")

Table 2. Search Results

Year	Database	Total Count
2000–2017	ACM Digital Library (proceedings, journals, newsletters, and magazines)	1,289
2001–2017	IEEE Xplore (conferences, early access articles, journals, magazines, and books)	1,214
2000–2017	SpringerLink (from sub-discipline "Information Systems Applications incl. Internet": chapters, conference papers, articles, and books)	3,196
2000–2017	ScienceDirect (reviews, research articles, and books)	4,096
1997–2018	Google Scholar (research articles, books, reports, and theses)	53
<i>Manual Search</i>		
	Total	9,848
	<i>Total (after screening)</i>	9,418

2.1.3 Search Process. During the search process, we found that many of the databases indexed each others' articles, therefore the chances of getting redundant results were high. Thus, to avoid duplicate results, we manually selected different options (like Publication Type, Publisher, etc.) while searching through each database. In total 9,418 articles were identified after removing 430 redundant articles. Table 2 presents the number of relevant articles identified from each academic database.

2.2 Study Selection

To identify the articles relevant to our research questions, we applied a two-phase selection process. During this process, two researchers of this review independently analyzed the identified articles and selected the studies, which were most likely related to our research questions.

2.2.1 Selection Phase 1. In this phase, we studied the titles and abstracts of the identified articles and assessed them on the basis of the inclusion criteria listed in Table 3. After completion of this phase, 216 primary studies (PS) were selected. We then scanned the reference list of the selected primary studies to identify related articles that we might have missed during our initial search. We found three articles which passed our inclusion criteria, and therefore we added these articles to our list of primary studies; making a total of 219 articles (see Table 2 in Supplementary Material).

2.2.2 Selection Phase 2. In this phase, we applied the quality assessment criteria illustrated in Table 4 to the primary studies selected in Selection Phase 1. After completion of this selection

Table 3. Inclusion Criteria for Selection Phase 1

Criteria ID	Inclusion Criteria
IC1	The article explains the theoretical foundations of collective intelligence in computer science.
IC2	The article describes the role of collective intelligence in crowdsourcing and open innovation.
IC3	The article focuses on architecture/frameworks of CI systems.
IC4	The article describes CI systems/applications available on the Web.
IC5	The article focuses on knowledge generation and exchange in crowds.
IC6	The article is related to at least one aspect of our research questions.
IC7	The article should not compare collective intelligence with swarm intelligence and artificial intelligence.

Table 4. Quality Assessment Criteria for Selection Phase 2

Criteria ID	Quality Criteria Check-List
QC1	Are the research objectives clearly defined in the study?
QC2	Does the study propose a new framework, or provide technological details of an existing CI system?
QC3	Is the system architecture/framework/design/experiment clearly defined in the study?
QC4	Is the proposed CI architecture or framework compared to existing CI models or systems?
QC5	Does the study provide insights about the role, importance, and behavior of individuals in the proposed CI system or model?
QC6	Does the study propose novel solutions to crowd management issues in CI?

phase, 12 primary studies were finally selected. These 12 studies were then used for data extraction and data synthesis. We describe both stages further in the next sections.

2.3 Study Quality Assessment

The intention of this phase is to determine the relevance of selected studies while limiting bias in the study selection process. In this phase, all three researchers of this review independently assessed the primary studies by answering the questions presented in Table 4. For each primary study, the researchers answered the questions as “Yes,” “Partly,” or “No”; scoring each criteria as 1, 0.5, and 0, respectively. The individual scores for each question were then added to derive a total score for each primary study. The studies that scored 3 or higher were finally selected for the data synthesis stage. Any conflict of opinion about the process and results of the quality assessment measures were discussed among all three researchers to reach a consensus. The scores of the remaining 12 primary studies that satisfied the quality assessment criteria are presented in Table 5; followed by the title of the studies and the publication type presented in Table 6.

2.4 Data Extraction and Synthesis

The intention of data extraction stage is to identify the main contributions of the selected studies, and to present a summary of the work. Table 7 presents the data items extracted from the 12

Table 5. Quality Score of Selected Studies

Primary Study ID	QC1	QC2	QC3	QC4	QC5	QC6	Total Score	Selected Study ID
PS1	1	1	0.5	0	0.5	1	4	S1
PS19	1	0	0.5	0	1	1	3.5	S2
PS48	1	1	0.5	0.5	0.5	0	3.5	S3
PS73	1	1	1	1	1	0	5	S4
PS108	1	1	0.5	0	0.5	0	3	S5
PS138	1	0	1	1	0.5	0	3.5	S6
PS154	1	0	1	1	0	0	3	S7
PS155	1	0	1	1	0	0	3	S8
PS156	1	1	1	1	0	0	4	S9
PS173	1	1	1	1	1	0	5	S10
PS174	1	0.5	0.5	0	0.5	0.5	3	S11
PS204	1	1	1	1	0	1	5	S12

Table 6. List of Final Selected Studies

Study ID	Study Title	Publication Type
S1	“Intelligent Collectives: Theory, Applications and Research Challenges” [58]	Journal Article
S2	“Leadership and the Wisdom of Crowds: How to Tap into the Collective Intelligence of an Organization” [53]	Journal Article
S3	“Modelling the Index of Collective Intelligence in Online Community Projects” [73]	Conference Paper
S4	“The Role of Collective Intelligence in Crowdsourcing Innovation” [67]	PhD Thesis
S5	“Collective Intelligence Model: How to Describe Collective Intelligence” [21]	Conference Paper
S6	“Collective Intelligence Systems: Classification and Modeling” [48]	Journal Article
S7	“Designing for Collective Intelligence” [27]	Journal Article
S8	“On Model Design for Simulation of Collective Intelligence” [70]	Journal Article
S9	“A Resource Allocation Framework for Collective Intelligence System Engineering” [81]	Conference Paper
S10	“Harnessing Crowds: Mapping the Genome of Collective Intelligence” [52]	Journal Article
S11	“Leveraging the Power of Collective Intelligence through IT-enabled Global Collaboration” [32]	Journal Article
S12	“Collective Intelligence: A Keystone in Knowledge Management” [3]	Journal Article

selected studies and Section 3 presents a summary of the same. The contributions, i.e., models and elements of the Selected Studies, are presented in Table 16.

The goal of the data synthesis stage is to collate and summarize the contributions of the selected studies. In Section 4, we first catalog the definition types and classifications of the studied CI models; we then identify all unique and synonymous *characteristics*, *levels*, *requirements*, *properties*, and *building blocks* and classify them into 24 distinct attributes (presented in Table 16). Finally,

Table 7. Data Extracted from Selected Studies

Extracted Data Item	Description
Study Title	See Table 6.
Author(s)	See Table 2 in Supplementary Material.
Year	See Table 2 in Supplementary Material.
Publication Title	See Table 2 in Supplementary Material.
Publication Type	See Table 6.
Source/Publisher	See Table 2 in Supplementary Material.
Summary	See Section 3.

Table 8. Criteria for Collective to be Intelligent (as Presented by Nguyen et al. in S1) [58]

Criteria	Description
Diversity	Individuals must belong to diverse backgrounds, knowledge bases, and so forth.
Independence	Freedom for individuals to act according to their choice, without others influence.
Decentralization	Facilitate individualism and assure diversity in individuals.
Aggregation	Appropriate methods to integrate individual solutions [58].

based on these findings, we then answer the first two research questions (RQ1 and RQ2) in Section 4 and the final research question (RQ3) in Section 5.

3 SUMMARY OF SELECTED STUDIES

3.1 S1 (Van Du Nguyen et al. 2018)

The aim of this study is to define the criteria necessary for a collective to be intelligent. To do so, the study [58] presents a novel general CI framework based on crucial attributes of a collective.

Influenced by Bonabeau’s [4] concept of Decisions 2.0, which is defined as “a new era of decision-making in which the traditional decision-making process is supported using the wisdom of crowds through collaboration and collective intelligence” [58], Nguyen et al. state that “collective intelligence is considered as the power of Decisions 2.0” [58]. Based on this premise, the study proposes a CI framework based on characteristics vital for an intelligent collective, as proposed by Surowiecki [77]. According to Nguyen et al., a collective must fulfill four criteria (presented in Table 8) [58] to be intelligent. And, based on these characteristics, the authors propose a general framework of CI (namely, *Collective, Aggregation Methods*, and *Collective Performance Measures*) [58] for wisdom of crowds.

3.1.1 Diversity. A collective must be diverse, as a heterogeneous group of individuals can provide new knowledge and diverse viewpoints to any given problem. Nguyen et al. further categorize diversity as “diversity in the composition of collective members” [58] and “diversity of individual predictions in a collective” [58]. To explain diversity, the authors used an example of weather forecasting; where accurate weather predictions are a difficult task even if relying on experts. The authors claim that such prediction problems could be solved more easily if multiple individuals were allowed to add extra information and provide different perspectives to solve the problem [58].

3.1.2 Independence. The individuals in a CI system must be allowed to provide their own inputs and their decisions should not be influenced by others [58, 77]. This is important, because information cascades can diminish the intelligence of the collective [1].

3.1.3 Decentralization. This criteria helps individuals act independently, while avoiding others' influence, and thus ensures diversity [58]. To explain this, the authors used the example of Linux, where solutions to specific problems are selected from a pool of solutions submitted by independent programmers from around the world.

3.1.4 Aggregation. This criteria provides the appropriate mechanism to integrate the opinions and solutions provided by the individuals [58]. Examples of such new aggregation methods include prediction markets [82] and social tagging [86].

3.2 S2 (Kurt Matzler et al. 2016)

The aim of this study is to present activities necessary to promote collective intelligence within organizations. The proposed activities are based on the work of Surowiecki [77] and are explained using case studies and real world examples [53].

In this study, Matzler et al. argue that although platforms such as wikis, blogs, prediction markets, and so forth, might be enough to harness the wisdom of the crowd from end users, such platforms are inadequate to support collective intelligence within organizations. The authors propose that in order to harness the power of collective intelligence within organizations, it is imperative that managers follow the following steps: "create cognitive diversity" [53], "promote independence" [53], "access decentralized knowledge" [53], and "effectively aggregate knowledge" [53].

3.2.1 Cognitive Diversity. To explain cognitive diversity, Matzler et al. refer to the work done by Page [61]. Page states that cognitive diversity can be explained as a combination of "diverse perspectives" [53], "diverse interpretations" [53], "diverse heuristics" [53], and "diverse predictive methods" [53]. Matzler et al. explain the relevance of these attributes in organizations by the means of two case studies; namely, "How diversity can drive innovation" [30] and "The CEO's role in business model reinvention" (a case study from Infosys Technologies Limited) [24].

3.2.2 Promote Independence. Matzler et al. emphasize the importance of this step, by explaining how lack of independence or peer pressure may force employees to convey incorrect or sugar-coated information to their managers, which may lead to biased decisions [53]. The authors suggest that managers should create an atmosphere of open dialog where all employees can share their honest opinions and ideas; the authors recommend techniques like the PreMortem exercise [38] to create such an independent environment.

3.2.3 Access Decentralized Knowledge. In regard to this step, Matzler et al. state how, in the past, knowledge was organized hierarchically in organizations; where as now, due to globalization, decentralization, and data ubiquity, knowledge within organizations is not limited to the organizations themselves [53]. In other words, when looking for novel solutions and ideas, organizations now rely heavily on participants via online contests, social media platforms, blogs, and wikis [74]. Matzler et al. argue that organizations could boost their internal collective intelligence by allowing their employees to tap into this decentralized knowledge aggregated from the social web [53]. Employees could then use this knowledge to come up with ideas and solutions to support the organization's growth, while being aligned with the organization's vision and mission.

3.2.4 Effectively Aggregate Knowledge. The final step for promoting collective intelligence within organizations is to effectively aggregate dispersed knowledge. In this study, Matzler et al. briefly discuss techniques (such as averaging individual opinions) that could be utilized to aggregate knowledge from different sources [53]. The authors further describe this step using the examples of predictive markets and peer review systems, which have been found to be effective knowledge aggregation techniques [29]. Lastly, Matzler et al. discuss the effectiveness of Wikipedia's peer review system [53], by comparing the accuracy of its knowledge base to that of Britannica, as investigated by Jim Giles [22].

Table 9. Levels for Assessing CI Potential (as Presented by Skarzauskiene et al. in S3) [73]

Level	Description
Capacity Level	Describes possible user actions, both as an individual and as a member of the community [48]. It also includes massive participant interactions [47] that promote knowledge creation and innovation [3].
Emergence Level	Describes a system state [48] that supports self-organizing, “emergent” behavior, “swarm effect” [47], and mechanic development [3].
Social Maturity Level	Describes the clarity of system goals [3] and community/individual objectives [48].

3.3 S3 (Aelita Skarzauskiene et al. 2015)

The aim of this study is to propose measures to quantify the minimum potential required by community projects, necessary to transform them into CI systems. The authors do so by investigating the trends in engagement and participation in online communities in Lithuania. Skarzauskiene et al. conduct both qualitative and quantitative research by extensively interviewing 20 individuals and by conducting a public opinion survey with 1,022 Lithuanian participants between the ages of 15 and 74 [73]. Finally, the authors propose three levels/measures a community project must fulfill in order to be considered as a CI system [73].

Before conducting qualitative and quantitative research, Skarzauskiene et al. briefly analyzed several CI frameworks proposed by researchers. Based on the literature, Skarzauskiene et al. proposed a conceptual framework for assessing the potential of CI [73]. The authors define the proposed conceptual framework in three levels, presented in Table 9 [73]. Using the proposed levels, combined with results of qualitative and quantitative analysis, the authors calculate a CI Potential Index, which they claim could assist developers and initiators of community projects by helping to assess the CI potential of such projects [73].

3.4 S4 (Juho Salminen 2015)

The aim of this doctoral thesis is to explore the role of collective intelligence in crowdsourcing innovations [67]. Salminen’s work is motivated by the fuzzy nature of CI, which has led to different interpretations of the concept including “wisdom of crowds” [77] and “swarm intelligence” [5]. In his work, Salminen attempts to defuzzify the notion of collective intelligence by investigating its emergence as a complex-adaptive system [67].

To do so, the author conducted a systematic literature review of published case studies discussing three CI platforms (OpenIDEO, Quirky, and Threadless). He then observed user behavior on each of these platforms for over a month, and gathered relevant data including web clips, diary entries, and statistics. Salminen also conducted a literature review of available CI frameworks, based on which he proposed a new theoretical framework for CI. Finally, he evaluated the proposed framework based on his observations from the previously analyzed CI platforms. Salminen defines the proposed framework through three levels of abstraction [67]:

- *Micro*: “enabling factors of collective intelligence.”
- *Emergence*: “from local to global.”
- *Macro*: “output of the system and wisdom of crowds” [67].

Table 10 presents the elements of Salminen’s proposed theoretical framework based on themes from literature. Apart from the proposed theoretical CI framework, Salminen also highlights the crucial issue of biased feedback. When observing the previously mentioned CI platforms, the author found that participants would often create multiple accounts to vote up their own ideas and

Table 10. Themes and Elements of the CI Theoretical Framework (as Presented by Salminen in S4) [67]

Level	Theme	Elements of the Theoretical Framework	References
Micro	Humans as social animals	Human capabilities for interaction *	
	Intelligence		
	Personal interaction capabilities		[12, 35, 85]
	Trust		[2]
	Motivation		
	Attention		
	Communities [67]		[8, 10]
Emergence	Complex adaptive systems	Agents, activities, feedback, emergence	[60]
	Self-organization	Agents, activities, feedback	[36]
	Emergence	Emergence	[14]
	Swarm intelligence	–	
	Stigmergy	Agents, activities, feedback, distributed memory	[78]
	Distributed memory [67]	Distributed memory	[6]
Macro	Decision-making	Output	
	Wisdom of crowds	Output	[77]
	Aggregation	–	
	Bias	–	
	Diversity	–	[31]
	Independence [67]	–	

* Same for all themes in micro level.

would demoralize their competitors by providing negative/incorrect feedback and down-votes. Salminen states that to prevent such issues, researchers must create measures to evaluate the accuracy of crowd decisions [67].

3.5 S5 (Sandro Georgi et al. 2012)

The aim of this study is to build a comprehensive model based on available literature while recognizing the characteristics that describe CI [21].

Georgi et al. draw attention to a very important issue in the field of CI, i.e., that research about the topic in general is very limited, as most available research is application- and type-specific. The authors state that although numerous scientific articles and reports have been published about CI platforms, frameworks, and models, only little research has been done on “how to describe collective intelligence in general” [21]. To fill this gap, the authors first studied the existing scientific literature and choose three models of CI, namely, “the collective intelligence genome” by Malone et al. [52], “mitigating biases in decision tasks” by Bonabeau [4], and “the collective intelligent system” by Lykourantzou et al. [49]. Combining these three models, the authors propose five novel characteristics and argue that these can appropriately describe collective intelligence. Table 11 presents these characteristics and their descriptions as stated by Georgi et al. [21].

3.6 S6 (Ioanna Lykourantzou et al. 2011)

This study aims to design a modeling process that can identify the common characteristics of CI systems. Additionally, the process helps to identify challenges that prevent the construction of a generic CI system [48].

Lykourantzou et al. claim that their work is the first attempt in classifying the common shared characteristics of CI systems. The authors state that although all CI systems may seem to be

Table 11. Characteristics that Define CI (as Presented by Georgi et al. in S5) [21]

Characteristics	Description
Objective of task	Can be described as the outcome that the CI intends to achieve. These objectives can be categorized as “create” (creation of knowledge or ideas or physical objects) and “decide” (correctness or best or most suitable, respectively).
Size of contribution	Represents the amount or volume of contribution, and can vary depending upon the complexity of the problem and form/structure of the CI.
Form of input	Can be presented in the form of rules or data/information (pictures, text, datasets, etc.), and can be categorized as instructions, challenge descriptions, or raw material.
Form of output	Can be of two types: knowledge (i.e., intangible) or products (i.e., tangible).
Stakeholder	Defines stakeholders of a CI system based on their roles. “Initiators” are those whose objective is to reach a desired goal. “Contributors” do the actual work and use their intelligence to provide solutions. Finally, “beneficiaries” are those who profit from the outcomes of such systems [21].

Table 12. Common Characteristics that Define a CI System (as Presented by Lykourantzou et al. in S6) [48]

Characteristics	Description
Set of possible individual actions	Set of actions that an individual is allowed to perform when contributing (in some form or another) within the system.
System state	Set of minimal variables that completely define the system.
Community and individual objectives	List of goals that a community or an individual intends to achieve by using the system.
Expected user action function	Effort expected from users, necessary to achieve individual/community goals.
Future system state function	Expected state of the system after some time, given the system’s current state and user actions.
Objective function	Measures the extent to which individual/community goals of the system have been achieved [48, 81].

substantially different from each other, they all seem to share quite a few characteristics. After analyzing published literature on CI, Lykourantzou et al. proposed that all CI systems could be categorized as either “active” or “passive” systems. Additionally, “active” CI systems could further be classified into “collaborative,” “competitive,” or “hybrid” systems [48]. The authors suggest that in “passive” CI systems, groups of users would exhibit behavior of swarms, irrespective of whether the system requires such a behavior or not. Whereas in “active” systems, crowd behavior is created and coordinated by the system itself [48].

Lykourantzou et al. further state that based on this classification, CI systems have several common attributes (described in Table 12) [48]. The authors also highlight issues of “critical mass,” “task and workload allocation,” and “motivation” that should be considered when designing CI systems. Finally, Lykourantzou et al. model three types of CI systems (*Collaborative*: Wikipedia and open source software development communities; *Competitive*: Innocentive, BootB, DesginBay,

Table 13. Requirements for CI Applications (as Presented by Gregg in S7) [27]

Requirements	Description
Task-specific representation	CI applications should support task-specific views depending upon the application domain.
Data is the key	The effectiveness of CI applications is directly proportional to its data quality and quantity, and therefore should facilitate data collection and sharing among its users.
Users add value	CI applications should help users to improve the usefulness of data, by providing mechanisms that enable user-oriented addition, modification, or enhancement of data.
Facilitate data aggregation	Keeping the importance of data in mind, CI applications should be designed with necessary features that enable data aggregation throughout the duration of the systems' use.
Facilitate data access	CI applications should offer services and mechanism that facilitate reuse of data outside the application.
Facilitate access for all devices	CI applications should provide services that are usable not just with PCs and internet servers, but also portable devices like PDAs, smart-phones, and tablets.
The perpetual beta	New features must be added to CI applications from time to time, depending upon the community needs and requirements [27].

DARPA Network Challenge; *Passive*: vehicular ad-hoc networks) using the previously identified attributes [48].

3.7 S7 (Dawn G. Gregg 2010)

The aim of this study is to demonstrate the requirements for designing CI applications. Gregg states that a CI application harnesses the knowledge of its users by facilitating human interaction and decision-making, and therefore, new CI applications must center around the importance and use of user-defined data [27]. Inspired by the work of O'Reilly [79], Gregg proposes seven key requirements for CI applications (described in Table 13) [27].

To illustrate how these requirements could be used to design CI applications, the author developed the "DDtrac" application for children with special needs. The application was intended to support decision-making in special education and therapy. DDtrac is a web-based CI application and has two main objectives: first, the application facilitates communication between therapists and teachers so that they could share information about the needs of the children; second, the application allows data collection and provides tools for data analysis to understand a child's progress and to determine adjustments necessary for a better development of the child. The application was deployed for a duration of 18 months with one autistic student and his teachers and therapists. After the conclusion of the trial, all participants reported that the application successfully achieved both its core objectives and helped to improve the academic performance of the student [27].

3.8 S8 (Martijn C. Schut 2010)

This study aims to provide systematic guidelines and instructions for development of CI models, irrespective of the developer's domain. To come up with these guidelines the author first conducted a number of research studies and identified key contributions which distinguish CI systems from other ICT systems. Based on the literature, Schut compiled a list of properties of CI

Table 14. Properties of CI Systems (as Presented by Schut in S8) [70]

Types	Properties
Enabling CI properties	<p data-bbox="427 295 1192 352">These properties enable the emergence of collective intelligence in a system.</p> <ul data-bbox="475 382 1192 662" style="list-style-type: none"> <li data-bbox="475 382 1192 439">• <i>Adaptivity</i> refers to the capability of a system to change its behavior or structure depending upon the environment. <li data-bbox="475 445 1192 567">• To understand system behavior, it is important to understand both individual actions and <i>interactions</i> among individuals as a whole. These interactions enable the flow of information within systems. <li data-bbox="475 573 1192 662">• Individual or system behavior can be described fundamentally using <i>rules</i>. Such rules implicitly represent the relationship between system inputs and outputs [70].
Defining CI properties	<p data-bbox="427 696 1192 725">If these properties exist in a system, it can be considered a CI system.</p> <ul data-bbox="475 755 1192 1287" style="list-style-type: none"> <li data-bbox="475 755 1192 906">• <i>Global-Local</i> are levels which distinguish between aggregation at system and individual level, respectively. This distinction is important for understanding adaptivity and emergence. Adaptivity can occur at local and/or global level, whereas emergence is achieved by going from local to global. <li data-bbox="475 912 1192 969">• Complex systems must have elements of <i>randomness</i> in order to behave as self-organized critical systems. <li data-bbox="475 974 1192 1096">• <i>Emergence</i> is defined as the principle that “the whole is greater than the sum of its parts” [14], and occurs when moving from “the lowest abstraction level (individual) to the highest abstraction level (system)” [70]. <li data-bbox="475 1102 1192 1191">• <i>Redundancy</i> means that the system should allow its users to access/visualize available knowledge and information at different locations within the system’s user interface. <li data-bbox="475 1197 1192 1287">• Redundant data can make the system <i>robust</i>, as data that are lost due to malfunctions could still be recovered from other sources [70].

systems [70] presented in Table 14. After this, the author investigated several strands of research such as complex adaptive systems, swarm intelligence, and others, that are often described as being synonymous or at least associated to collective intelligence [70]. Based on the findings, Schut finally proposed a “systematic approach for designing CI system models” [70] and illustrated the proposed methodology using two case studies, namely, the “Chinese Whispering Room” and the “Braitenberg collectivae” [70].

The CI system modeling approach proposed by Schut is divided into three phases, i.e., “system design,” “model design,” and “models” (which are further categorized into “generic,” “system,” and “computer” models) [70]. The components of the “system design” phase are inspired by examples from self-organization, multi-agent systems, and swarm intelligence; whereas the components of the “model design” phase are influenced by the work of van Gigch [80] on system modeling and of meta-modeling.

3.9 S9 (Dimitrios J. Vergados et al. 2010)

The aim of this study is to present a framework that can foster the emergence of CI in web community based platforms. Based on published research, Vergados et al. describe a generic CI system as having three main components, i.e., “human community,” “machine intelligence,” and “system information” [81].

Vergados et al. argue that although the proposed CI framework may lead to the development of completely different CI systems, all systems would share a number of common characteristics [49]. The authors describe these characteristics as follows [81]:

- **System attributes** (same as described in Table 12)
 - *Set of possible individual actions*
 - *System state*
 - *Community and Individual objectives* [48, 81]
- **Functions** (same as described in Table 12)
 - *Expected community member action functions*
 - *Future system state functions*
 - *Objective functions* [48, 81]
- **Other elements**
 - *Resource allocation algorithms*: These algorithms define the required user actions (depending upon the system state) necessary to reach user/system goals and to maximize the usefulness of the system.
 - *Critical mass*: This indicates “the minimum number of users necessary for the system to function effectively” [81].
 - *Motivation*: A vital factor, important to improve the quantity and quality of user participation in a CI system. [81]

Finally, Vergados et al. evaluate the proposed framework by means of simulation where they analyze how the quality of Wikipedia articles could be improved, if the system was based on the proposed concepts. The authors compare the performance of their approach against the current approach used in Wikipedia by using the mathematical functions of the proposed framework. The authors claim that, based on their framework, a CI-enabled Wikipedia community could significantly improve the quality of articles, while reducing the time required for these articles to reach satisfactory quality [81].

3.10 S10 (Thomas W. Malone et al. 2009)

This study aims to propose a new framework that explains the underlying model of CI systems. To do so, Malone et al. examined 250 web-enabled CI systems; and based on their findings, identified the building blocks or “genes” (analogy adopted from biology) of CI [52]. The authors then classified these building blocks, using two pairs of fundamental questions [52], i.e.,

- “Who is performing the task? Why are they doing it?”
- “What is being done? How is it being done?” [52]

The answers to these questions with respect to *staffing*, *incentive*, *goal*, and *structure/process* were then proposed as the “genes” of CI systems [52]. Malone et al. state that different CI systems could be modeled using the combination and recombination of these building blocks. A brief overview of these “genes” is presented in Table 15 [52].

To explain these genes further, Malone et al. examined four web-enabled CI systems: *Linux*, *Wikipedia*, *InnoCentive*, and *Threadless*. Finally, the authors claim that the “sequences of genes” of

Table 15. Building Blocks of CI (as Presented by Malone et al. in S10) [52]

	Genes	Description
Who?	Hierarchy	In this gene, tasks are assigned to individuals or groups by someone in authority (similar to traditional hierarchical organizations).
	Crowd	In this gene, individuals within the group can indulge in activities if they choose to do so; and there is no authoritative figure [52].
Why?	Money	Financial gain can be a big motivator for individuals in markets and organizations.
	Love	In many situations, emotional states such as love, affection, passion, or simply interest could be a great motivator for participants.
	Glory	Recognition by competitors, colleagues, or general public is another important motivator [52].
What?	Create	In this gene, participants create something like a T-shirt design, a piece of code, or an innovative solution to a given problem.
	Decide	In this gene, participants evaluate and select items from a set of options; primarily, submitted by other participants [52].
How?	<i>Create</i>	
	Collection	This gene occurs when participants create solutions independently. A sub-type of this gene is the <i>contest gene</i> , which occurs when one or many contributions are recognized as best and are rewarded.
	Collaboration	This gene occurs when participants create solutions as a group, and the proposed solutions are interrelated or interdependent.
	<i>Decide</i>	
	Group	This gene occurs when “members of a crowd make a decision that applies to the crowd as a whole” [52]. Important variants of this gene include <i>voting</i> , <i>consensus</i> , <i>averaging</i> , and <i>prediction markets</i> .
	Individual	This gene occurs when members of the crowd make their own independent decisions, which might be influenced by other members but are not necessarily identical. Two important variants of this gene are <i>markets</i> and <i>social media</i> [52].

each of these systems could be combined into *genomes* that could help us to understand these CI systems better [52].

3.11 S11 (Luca Iandoli 2009)

The aim of this study is to provide a model for management of CI, and to raise issues that must be considered when designing CI systems. Iandoli argues that although there are several open issues in CI, all of these issues could be organized into two macro-areas, i.e., “management of collective intelligence” [32] and “design of collaborative tools” [32].

Iandoli states that online/virtual communities could be viewed as organizations and, therefore, could also be modeled as such. Based on this hypothesis, Iandoli et al. proposed five characteristics of online/virtual communities “when modelled as organizations” [33]:

- (1) “Clear goals and objectives” [32, 33] coherent with a predefined mission.
- (2) “A large number of participants” [32, 33] who can offer their time and efforts to achieve the system goals (by knowledge sharing, creation, and consensus activities) in return for incentives.

- (3) “A set of processes” [32, 33] that allow participants to develop, submit, or evaluate new ideas, artifacts, and decisions by collaborating with others.
- (4) “Rules” [32, 33] that govern how participants interact with the system and each another.
- (5) “Participant roles and responsibilities” [32, 33].

Iandoli further argues that even if virtual communities are modeled as organizations, such communities would still face major governance issues because of the many differences between virtual communities and real organizations. Three of these issues [32] are the following:

- *Attention governance*: This involves reducing the possibility of premature, incomplete, or biased decisions, caused due to the lack of correct and unbiased knowledge or due to peer pressure.
- *Participation governance*: The system must facilitate and support participation of large numbers of individuals from diverse backgrounds. Participants must be provided with suitable incentives to keep them inspired and motivated to share their information and knowledge, and to help achieve the system objectives in an unbiased fashion.
- *Community governance*: Appropriate rules must be established to enable smooth and stable interactions among participants and communities; the system should be organized hierarchically and individuals should be given clearly defined roles, responsibilities, and incentives [32].

Finally, Iandoli states that even if all the above-mentioned issues are resolved, there would still be two challenges, i.e., “designing proper visualization tools” [32] and “designing trust and reputation appraisal systems” [32] that would have to be dealt with, irrespective of the technologies used when designing such collaborative platforms [32].

3.12 S12 (Andre Boder 2006)

This study aims to establish a new model for CI in organizations. The model is inspired by Nonaka’s work on “The Knowledge-Creating Company” [59] and provides insights that enable transformation from tacit to explicit knowledge within and among organizations, from a collective intelligence perspective [3].

Pertaining to literature on knowledge management in organizations, Boder argues that the process of how organizational elements (such as individuals, their expertise, formal and informal networks, methods of communication, and implicit cultural norms) interact to enable knowledge creation, represents a form of CI. Based on this argument, Boder presents the building blocks of organizational collective intelligence [3]:

- Block A (*Development of competencies*), i.e., the first block “is the development of competencies” [3]. Although difficult to realize, organizations should aim to develop complementary competencies. This could possibly be achieved by human resource managers, who should identify individuals with different competencies gained from different situations; and once such individuals are identified, knowledge managers should bring them together so that their competencies complement each other. Doing so, organizations could take advantage of individual competencies and therefore create new knowledge.
- Block B (*Goal development*). The second block “is the development of a common representation of the goals” [3]. Although each group or department within the organization could have its own goals and objective, these goals and their representations should be aligned with the organizations overall objectives and should be coherent.

- Block C (*Mechanic development*). The third block “is the development and alignment of processes into mechanics of interactions between entities involved” [3], i.e., *organizations*. The formal and informal norms of the organization must be stated explicitly; additionally, employees should respect each others expectations and should trust each others competencies, because such a culture would enable smooth articulation when dealing with new problems or challenges [3].

To illustrate how these building blocks could be used in the process of building CI, Boder breaks down these actions into six groups, and describes six generic tools that could be utilized to apply these actions [3]. He then uses these tools and actions to describe three scenarios: “the value chain” [3], “co-integration of key competencies to achieve a critical medical mission” [3], and “innovative problem-solving” [3]. Finally, the author concludes by stating that organizations must create novelty to survive and evolve. And this is only possible if organizations build collective intelligence CI by combining the know-how of their employees and integrate organizational knowledge with partner organizations by “coordinating their respective value chains” [3].

4 DATA SYNTHESIS

In this section, we look at the different definitions and classifications of elements that describe a CI model, as proposed by the studies discussed in Section 3. Looking at all these elements, it is clear that different authors define CI systems using different terminologies such as *characteristics*, *levels*, *requirements*, *properties*, and *building blocks*; however, a deeper examination of these models proves that each of these definition types propose similar (if not the same) concepts. Similarly, many of the selected studies explain CI from different perspectives (such as CI in organizations, CI as self-organizing systems, and others); however, the characteristics of CI presented in these studies are very much alike. Table 16 presents the list of all characteristics proposed in the selected studies and classifies them into 24 *unique* attributes (described in Section 5) according to their definitions (described in Section 3). It is important to note here, that some of the selected studies have proposed combinations of characteristics from previous research; and therefore, are presented as combinations of attributes in Table 16.

Based on the findings of the data extraction and data synthesis stages, we now answer the first two research questions.

4.1 Research Question (RQ1)

What are the underlying models of existing CI systems? What are the common terminologies used to describe CI models? What are their components? And, how are these components associated to each other?

4.1.1 RQ1.1. What are the Underlying Models of CI Systems?

Literature shows that CI is a multidisciplinary field, drawing concepts and techniques from a number of different disciplines including computer science [23], organizations [25], social media [69], complexity sciences [70], and psychology [84]; therefore, different scholars have described CI from different perspectives. However, over the years only three definitions of CI have been widely adopted in ICT; two of which were proposed in this decade. The first formal definition of collective intelligence (in ICT) was proposed by Pierre Lévy (1997) [43], followed by Jerome C. Glenn (2013) [23] and Thomas W. Malone (2015) [50]. Although each of the definitions describes CI in its own distinct way, nevertheless, when examined together, the definitions express CI as having three main components, i.e., *individuals* (with data/information/knowledge); *coordination and collaboration activities* (according to a predefined set of rules); and *means/platform for real-time*

Table 16. Terminologies Used (in S1–S12) to Describe CI Systems and Attribute ID(s) of Their Respective Classifications

Study ID	Definition Type	Classification	Sub-classification	Attribute ID(s)
S1	Characteristics	Diversity		A1
		Independence		A2
		Decentralization		A1, A2
		Aggregation [58]		A16
S2	Steps	Cognitive diversity		A1
		Promote independence		A2
		Access decentralized knowledge		A17
		Effectively aggregate knowledge [53]		A16
S3	Levels	Capacity level	Set of possible individual actions	A18
			Massive participant interaction	A19
			Competencies development	A7
		Emergence level	System state	A20
			Self-organizing	A8
			Emergent behavior	A9
		Social maturity level	Mechanic development	A10
			Community and individual objectives	A11
			Goal development [73]	A12
			Humans as social animals	A13
			Personal interaction capabilities	A19
			Trust	A10
S4	Levels	Micro-level	Motivation	A3
			Attention	A19, A3
			Communities	A4
			Complex adaptive systems	A8, A9
			Self-organization	A8
		Level of emergence	Emergence	A9
			Swarm intelligence	A8, A9
			Stigmergy	A8, A9
			Distributed memory	A17
			Decision-making	A13
		Macro-level	Wisdom of crowd	A13
			Aggregation	A16
			Diversity	A1
			Independence [67]	A2

(Continued)

Table 16. Continued

Study ID	Definition Type	Classification	Sub-classification	Attribute ID(s)
S5	Characteristics	Objective of a task		A12
		Size of contribution		A5
		Form of input		A21
		Form of output		A21
		Stakeholder [21]		A4
S6	Characteristics	Set of possible individual actions		A18
		System state		A20
		Community and individual objectives		A11
		Critical mass		A5
		Task and workload allocation		A14
		Motivation [48]		A3
S7	Requirements	Task-specific representation		A22
		Data is the key		A23
		Users add value		A6
		Facilitate data aggregation		A16
		Facilitate data access		A17
		Facilitate access for all devices		A17
S8	Properties	The perpetual beta [27]		A8, A9
		Enabling CI properties	Adaptivity	A8, A9
			Interactions	A19
			Rules	A10, A21
		Defining CI properties	Global-local	A16
			Randomness	A8
			Emergence	A9
			Redundancy	A22
			Robustness [70]	A24
S9	Characteristics	System attributes	Community and individual objectives	A11
			Set of possible individual actions	A18
			System state	A20
		Other elements	Resource allocation algorithms	A14
			Critical mass	A5
			Motivation [81]	A3

(Continued)

Table 16. Continued

Study ID	Definition Type	Classification	Sub-classification	Attribute ID(s)
S10	Genes	Staffing	Crowd	A4
			Hierarchy	A4
		Incentive	Extrinsic motivation	A3
			Intrinsic motivation	A3
		Goal	Create	A13
			Decide	A13
		Structure/Process	Collection	A15
			Collaboration	A15
			Group Decision	A15
Individual Decision [52]	A15			
S11	Characteristics	Clear goals coherent with mission		A12
		Large number of motivated participants		A5, A3
		A set of processes		A15
		Rules		A10, A21
		Roles and responsibilities [32]		A18
S12	Building Blocks	Competencies development		A7
		Goal development		A12
		Mechanic development [3]		A10

communication (viz., hardware/software). When combined, these components enable intelligent behavior in groups or crowds.

Table 17 is the result of segregating all the characteristics defined in Section 3 in terms of the just discussed three main components of CI systems.

4.1.2 RQ1.2. What are the Common Terminologies Used to Describe CI Models?

As suggested in the selected studies, CI models have been described using terminologies such as *characteristics* (S1, S5, S6, S,9 and S11), *steps* (S2), *levels* (S3 and S4), *requirements* (S7), *properties* (S8), *genes* (S10), and *building blocks* (S12). And, each of these terminologies is further segregated into different classification and sub-classifications as described in Section 3. However, as mentioned in the previous sections, the terminologies used in these models describe similar concepts, and therefore can be classified into unique attributes as presented in Table 16.

4.1.3 RQ1.3. What are the Components of CI Models? And, How are These Components Associated to Each Other?

Typically the components of ICT systems are classified as data, hardware, software, information, procedures, and people. However, since the selected studies describe CI models by the means of their characteristics, these characteristics can be interpreted as the components of CI models. Based on the definitions of CI [23, 43, 50], we can segregate these characteristics/attributes and their relationship into the three main components of CI as described in Section 4.1.1 and presented in Table 17.

Table 17. Unique Attributes of CI (from S1to S12) Segregated According to the Components of CI

Component	Characteristics	Attr. ID	Study ID(s)
Individuals (with data, information, knowledge)	Diversity	A1	S1, S2, S4
	Independence	A2	S1, S2, S4
	Motivation	A3	S4, S6, S9, S10, S11
	Crowd	A4	S4, S5, S10
	Critical mass	A5	S5, S6, S9, S11
	Users add value	A6	S7
Coordination and collaboration activities (according to a predefined set of rules)	Competencies development	A7	S3, S12
	Self-organization	A8	S3, S4, S8
	Emergence	A9	S3, S4, S8
	Trust and respect	A10	S3, S4, S8, S11, S12
	Community and individual objectives	A11	S3, S6, S9
	Clear goals and objectives	A12	S3, S5, S11, S12
	Wisdom of crowd	A13	S4, S10
	Task and workload allocation	A14	S6, S9
	Set of processes	A15	S10, S11
Means for real-time communication (viz., hardware/software)	Aggregate knowledge	A16	S1, S2, S4, S7, S8
	Access to decentralized knowledge	A17	S2, S4, S7
	Roles and responsibilities	A18	S3, S6, S9, S11
	Massive interactions	A19	S3, S4, S8
	System state	A20	S3, S6, S9
	Predefined input/output types	A21	S5, S8, S11
	Task-specific representation	A22	S7, S8
	Data is key	A23	S7
	Robust	A24	S8

4.2 Research Question (RQ2)

Do any of the available CI models appropriately define all CI systems, irrespective of their applications? Can these models be used to create CI systems for novel challenges?

Comparing all characteristics of the studied CI models (see Table 16) with the components of CI systems (described in Section 4.1.1 and presented in Table 17), we see that none of the studied models have all 24 unique attributes, and therefore cannot define all CI systems completely. However, the existing models provide insights that can assist in planning when designing a CI system; and point out challenges that would have to be solved in order to achieve a robust and adaptive CI system. Most authors themselves state that their proposed CI models only describe collective intelligence in specific domains (S2, S3, S4, S9, S11, and S12), and suggest that further research and investigation is required to gain a better understanding of generic CI systems (S1, S3, S6, S8, S10, and S11). Therefore, although particular CI models can be used to define CI systems for specific domains, the same models might not be as useful when designing CI systems from other disciplines.

Furthermore, since the proposed models are evaluated using either quantitative/qualitative interviews (S3), or case studies (S4, S8), or examples from scenarios (S12), or simulations (S9), or applications/systems built based on the models (S6, S7, S10), it is not possible to identify a single model that can be used (in its current state) to design CI systems for novel challenges. For now, the most generic CI model available in literature is the one proposed by Malone et al. (S10) [21]; however, this highly cited and accepted model needs to be developed further for a deeper and more accurate understanding of CI [21, 52, 76].

5 A NOVEL FRAMEWORK FOR CI

Using the findings from the data extraction phase of the SLR, we now attempt to contribute to the available CI models by proposing a unified framework for CI by combining the 24 unique attributes (see Table 17) of CI models identified from studies S1–S12. The purpose of the proposed framework is to answer the final research question (RQ3) and provide additional insights and explanations that can help us better understand CI systems in general. In order to evaluate the proposed “generic” CI model, we will compare the model to multiple CI systems, each designed for a different objective and belonging to different disciplines (see Section 6).

RQ3. Can we somehow combine the available knowledge of CI models and systems to create a unified model that could define all CI systems?

We combine the knowledge of the CI models studied in this SLR, and propose a novel framework that describes CI systems in a fine-grained manner. We do so by comprehensively classifying all components of the studied CI models into 24 unique attributes (see Table 16), and then categorizing them into three sections:

- a “generic” model that defines all CI systems;
- additional requisites for CI systems; and
- CI as a complex adaptive system.

While taking inspiration from the building blocks for CI proposed by Malone et al. [52], combined with the findings from Section 4.1.1, we propose a model that describes CI systems by the means of *staff*, *process*, *goal*, and *motivation*. Designed as an extension to Malone’s concept of building blocks, the proposed generic model segregates the originally proposed *genes* into more fine-grained *types*; introduces a new classification, namely, *interactions*; and suggests vital *properties* for the *staff* and *goal* building blocks of the generic model. Finally, remaining attributes that could not be accommodated into the building blocks are aggregated into the additional requisites category.

5.1 A Generic Model for CI Systems

As mentioned in Section 3, Malone’s genome model for collective intelligence [52] is based on two pairs of questions: “*Who is performing the task? Why they are doing it?*” and “*What is being done? How is it being done?*” [52]. Based on these questions, the authors proposed the analogy of *genes* categorized as staffing, incentives, goal, and process. Each of these categories was subdivided into individual *genes* which, when combined, created the genome of CI systems. Drawing from the literature, we decided to move away from the concept of genes, and rather examine the proposed *genes* as *types*. Doing so, we realized that the available *genes* could be segregated into new types and sub-types. And, while some of the *genes* could be better understood as *interactions* between *types*, others could be explained as necessary *properties* inherent to these new *types*.

5.1.1 Who is Performing the Task? The *staff* in CI are the actors who perform different tasks within the system (as suggested in S10). As literature suggests, these actors or individuals must interact with each other based on certain rules depending upon the structure (hierarchical/non-hierarchical) of the system. And, when viewed as a collective, the *staff* of a CI system must exhibit a specific set of properties for the system to function effectively.

- *Types*: The actors or individuals in a collective are the first component of a CI system, and therefore play a vital role in describing how the system functions (A6). Typically, these actors (A4) can be segregated on the basis of their roles and responsibilities (A18) within

the system. Drawing insights from S4, S5, and S10, we determine that actors in a CI system can be classified as follows:

- *Passive actors* or beneficiaries are individuals who aim to gain from the outputs produced by the CI system, but do not wish to contribute in the problem solving process. These beneficiaries could either be stakeholders who are financially motivated, or end-users who simply want to exploit the knowledge produced by the system (but do not wish to actively contribute). Examples of stakeholders in CI systems could be seen in the following projects: Threadless, InnoCentive [15], and GoldCorp [83]. Here, the host organizations crowdsource their problems (designing of T-shirts, research and development, and identifying ideal mining locations, respectively) to the general public, with the intention of using the produced knowledge or artifacts for their own advantage.
- *Active actors* or contributors are individuals who are involved in CI processes (defined in “How”); such actors use their knowledge and expertise and help to create innovative solutions to the given problem. Such contributors can be further divided into two categories, namely, *crowd* and *hierarchy*.
 - * *Crowd* in a CI system comprises actors who actively contribute new knowledge, information, or artifacts to the system. Such actors are allowed to carry out a predefined set of actions, based on concrete sets of rules and regulations; however, there is no authoritative figure that has direct control over the actors’ individual actions. Examples of crowd in CI systems can be seen in the following projects: Climate CoLab [34], WikiCrimes [19, 68], and WeKnowIt [42] where users contribute data and information about the weather, crimes, and disasters, and also help verify the authenticity of the accumulated knowledge. Whereas, in projects such as Threadless [15], members of the crowd contribute by creating new artifacts and deciding on the best.
 - * *Hierarchy* in a CI system comprises administrators and experts who are responsible for allocating tasks to the crowd. While the administrators monitor crowd behavior in the system and make sure that the community and individual goals of the collective are achieved, the experts analyze and verify the contributions of the crowd. Additionally, in some cases the experts also help in identifying the best contributions or solutions. An ideal example of such a hierarchy can be seen in WikiCrimes: institutional agents, monitor agents, reputation agents, and others are responsible for different administrative activities within the system [19, 68].
- *Properties*: To ensure that a collective exhibits intelligent behavior, the collective of actors in a system must have a few crucial properties. According to S1, S2, and S4, a CI system must promote diversity and independence among its actors, as this can enable the creation of novel solutions exploiting knowledge from individuals familiar with multiple domains and with different experiences. Also, these actors should be allowed to act independently, as this can help to get rid of peer pressure, and therefore to reduce user-generated bias. Finally, to enable an effective collective intelligence, a collective must have critical mass or a minimum number of actors as suggested in S5, S6, S9, and S11.
- *Diversity (A1)* in CI systems refers to the heterogeneous nature of actors, who belong to different age groups, genders, and educational, financial, and cultural backgrounds. This is important as, such diverse actors can provide diverse pieces of knowledge, perspectives, interpretations, and experiences; and this could lead to the creation of innovative solutions and better decisions. An example of the advantages of diversity in actors can be seen in InnoCentive [15]: organizations with small R&D groups crowdsource their problems to acquire new and innovative ideas.

- *Independence* (A2) means that the opinions of one actor should not be influenced by the opinions of others. Independence among actors is vital, as it can help to avoid information cascades where users pass information that they assume to be true (without appropriate evidence or knowledge), and therefore make irrational choices and decisions [1, 46, 55].
- *Critical mass* (A5) in collectives is defined as the minimum number of actors who must participate in system processes for the system to function effectively. Although studies suggest that critical mass is an imperative property that enables effective creation and constant exchange of diverse knowledge and information, the concept needs to be investigated further as critical mass in different CI systems can often depend upon the system goals and objectives.
- *Interactions*: Interactions in CI systems can either exist between two or more actors, or among actors and the contributions of others. Such interactions can be categorized as follows:
 - *Trust and respect* (A10) are two preconditions for cooperation. When dealing with new problems or challenges, actors in a collective must treat each other with respect and should trust each others’ abilities and competencies, as doing so can enable smooth and efficient flow of knowledge and information within the system.
 - *SECI*: “Socialization, Externalization, Combination, and Internalization” [59] (A7) are the four components of Nonaka’s model for knowledge creation in organizations [59]. Using these knowledge dimensions, organizations can convert their employees’ tacit knowledge into explicit organizational knowledge and back. Since the SECI model was originally designed to promote sustainable innovation in organization, these concepts can also be utilized in CI systems to enable competency development in actors (as suggested in S12 and [11]).

Finally, as suggested in S3, S4, and S8, a CI system must support such interactions in massive volumes (A19).

5.1.2 Why They are Doing It? Motivation (A3) in CI systems is essential to maintain user engagement and encourage participation. Depending upon on the objectives of a system, users in a CI system could be motivated by their desire to gain knowledge (as in Wikipedia [45]) by money and glory (as in Threadless [8] and InnoCentive [57]) or by social cause (as in hackAIR [54]). According to Malone et al. (S10), money, love, and glory can be considered high-level motivations for people participating in CI systems [52]; whereas, Vergados et al. (S9) categorize motivation as tangible, intrinsic, and self-fulfilling [81]. Combining the recommendations from S4, S6, S9, S10, and S11, we categorize motivation as *intrinsic* and *extrinsic*.

- *Intrinsic* motivations such as social cause, interest, passion, and self-fulfillment encourage actors in a collective to collaborate and contribute for the betterment of the community or its individuals. An example of such motivation can be seen in DDtrac: school teachers and therapists collaborate to understand a child’s needs and determine necessary adjustments in teaching techniques for better development of children with special needs [26, 27].
- *Extrinsic* motivations are factors external to CI tasks that encourage actors to contribute in hopes of getting rewards. Such motivations can be either *tangible* like money and trophies or, *intangible* like fame and glory. In CI projects like Threadless [8], InnoCentive [57], and Goldcorp [83] participants are offered cash rewards and prizes for submitting ideas and designs; whereas, in WikiCrimes [19] participants gain a reputation based on the reliability their contributions.

5.1.3 What is Being Accomplished? Unlike Malone’s gene model (S10) that attempts to answer the question of “What is being done?” (from an organizational perspective), we decided to focus on the question of “What is being accomplished?” for our proposed model. Based on the literature, we found that our question is a better fit, as it could appropriately define the different types of objectives/goals (of CI systems) presented as characteristics in several selected studies. In general, these *goals* can be defined as “observable and measurable desired results bound to one or more objectives, that have to be achieved by committed actors within a finite time-frame.” Since collective intelligence initiatives are typically motivated by *community* or *individual* objectives (A11) as suggested by S3, S6, and S9, we segregate CI goals into the two aforementioned *types*. These types can be seen again in Threadless: individuals with a niche in T-shirt designing participate in competitions to present their contributions to the community, learn from others’ feedback, and earn money; whereas, the community’s goal is to bring new T-shirt designs to the marketplace by choosing and popularizing trending designs [8]. Additionally, drawing from the contributions of S3, S5, S11, and S12 the requisite properties of these CI system goals could be categorized as *well-defined* and *objective* (A12).

5.1.4 How is It Being Done? Malone et al. categorized the processes in CI systems as combinations of dependent-independent and create-decide activities, where the create-decide activities answered the question “What is being done?” [52]. In our proposed model, however, we describe CI *processes* (A15) as *types* of activities and *interactions*. As literature suggests, the activities can be either *create*, where actors come up with new ideas or design new artifacts; or it can be *decide*, where actors express their likes or dislikes for a particular subject or artifact. Since both of these activities can be either be done by individual actors or groups of actors, these activities could also be viewed as *dependent* or *independent* interactions. To add more granularity to process types, create activities can be further classified into *contest* (S10) and *voluntary*. As the names suggest, *contest* create activities are carried out in competitive environments and are extrinsically motivated, whereas *voluntary* create activities are intrinsically motivated. It is the combination of these three types (decide, contest, and voluntary) and interactions (dependent and independent) that enables intelligence in collectives (A13).

- *Collection* (i.e., *create* plus *independent*): In such activities or processes, actors participate as individuals and their contribution to the system is a result of their independent work. An example of *collection through contest* can again be seen in Threadless: individuals compete for cash rewards by creating and submitting new T-shirt designs [8]. Whereas, in Wiki-Crimes, actors contribute through *voluntary collection* by reporting criminal activities they witness in their local vicinity [19].
- *Collaboration* (i.e., *create* plus *dependent*): Such activities are carried out by groups of actors or communities where multiple individuals work together as a single entity and create new ideas or products.
As an instance for *voluntary collaboration*, we can again look at DDtrac: therapists and teachers work together to maximize the learning outcomes of students with special needs [27]. Similarly, in hackAIR, volunteers from NGOs conduct workshops to build citizen interest in the hackAIR platform and educate them on how they could become a part of the project’s community and help to gather air quality data from their local vicinity [40]. Whereas, *collaboration in contests* is seen in openIDEO, where multiple participants work as a team and propose solutions to societal challenges, in hopes of getting financial rewards [67].
- *Individual decision* (i.e., *decide* plus *independent*): Such decisions are made by individuals acting as independent entities and can be different for different actors. However, in some

cases these decisions may be influenced by the information provided by other actors. For instance, in Threadless the members of the community independently vote for T-shirt designs submitted by the participants. Unfortunately, as suggested by Salminen in S4, in some cases participants tend to create multiple accounts with the intention to down-vote their competitors, thereby influencing other members and generating biased community feedback [67].

- *Group decision* (i.e., *decide plus dependent*): In such activities, decisions are made by multiple individuals as a group or a community, and the outcome of the decisions impacts the community as a whole. For instance, such consensus can be seen in Threadless: the employees of the organization review the T-shirt designs chosen by the community and finally decide which designs to produce and award [8].

5.1.5 Input and Output. The final component of a CI system is the *flow of information*, or form of input/output (A21), and can be explained as interactions between the “who” and the “how” of the system. The flow of information starts from the *actors* who are responsible for providing inputs like individual knowledge and experiences, data from sensors, or end-user opinions and feedback from social media platforms. The collected inputs are then processed using different activities in “how,” and the results of these activities are then presented back to the *actors* who now take new decisions or produce new artifacts based on this newfound knowledge. Since this flow of information between the *actors* and the *processes* of the CI system is so vital, we decided to add it to our generic CI model.

The aggregation of the aforementioned components is illustrated as the proposed “generic” model for CI systems, in Figure 1.

5.2 Additional Requisites

Although any CI system can be described as the combination of the above-mentioned components, there are a few additional requisites that must exist in a CI system for the system to work effectively.

- *System state* (A20). This can be expressed as the minimum set of variables that completely define a CI system. As discussed in S3, S6, and S9 the system state can include challenges/issues raised by the members of the community, the identified solutions, activities of the users, and the system resources. Since our proposed model defines CI systems as the combination of different processes, actors, motivations, and goals, unique combinations of the same can be used to express the system state of a CI system.
- *Data is the key* (A23). “Collective intelligence draws on user-generated content and sharing of information, knowledge and ideas” [69] and, therefore, data or information/knowledge provided by members of the collective is a vital component of a CI system. For a CI system to be able to reach its goals, the system must allow its users to collect, manipulate, and share large volumes of data; this can enable robust innovations and decisions.
- *Aggregate knowledge* (A16). Since the effectiveness of a collective intelligence relies primarily on user-generated data/information, CI systems must have mechanisms and processes that aggregate this data/information. These aggregation processes are important as information provided by the community can often come from a variety of sources and could be incorrect or biased [20]. Aggregating the information, however, could help resolve conflicting information and could therefore allow for better innovations and reliable decisions. Additionally, systems should also provide mechanisms that allow users to aggregate their knowledge by means of social tagging (for information retrieval), collaboration (for exchange of vocabularies), and task-specific representation.

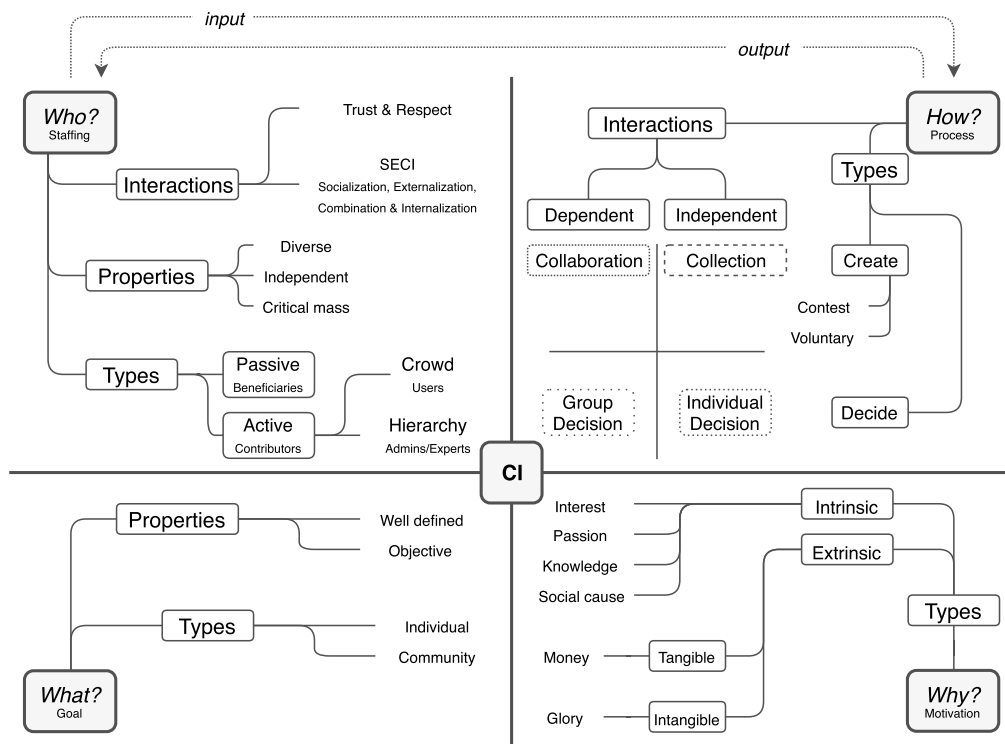


Fig. 1. Generic model for collective intelligence systems.

- *Access to decentralized knowledge (A17)*. Thanks to the growing number of internet users, more and more people are able to communicate, collaborate, and share information on the Web. Keeping user interest in mind, it is important for CI systems to allow users from different parts of the world to participate and gain from the knowledge or artifacts generated by the system. To do so, the system must facilitate access across multiple devices like PCs, laptops, smart phones, servers, and others. Furthermore, CI systems should support open data and open innovation practices, and should allow data access to users even outside the system.
- *Task and workload allocation (A14)*. Another important aspect that should be kept in mind when designing CI systems is the methods for coordination and resource allocation. When designing CI systems, the expected tasks of different actor types should be predefined; and based on these tasks, the rules and extent of interactions among actors and actor’s access to the aggregated knowledge must be outlined. For instance, participants should be allowed to add new solutions and view solutions submitted by others; however, they should not be allowed to make changes to others’ contributions without the contributor’s consent. On the other hand, system administrators should have complete access to the data/information/knowledge produced within the system.
- *Task-specific representation (A22)*. To support knowledge creation and enable fluid information exchange among actors from asynchronous groups, CI systems should provide task-specific representations like tables, charts, histograms, plots, and knowledge graphs. Additionally, depending on the task or problem, the system should allow its users to visualize the same knowledge/information in different forms.

- *Robust* (A24). Finally, since CI systems are designed as complex systems with multiple components, actors, users, and resources; it is important for such systems to be able to handle redundant and erroneous inputs. In addition to this, the system should also have appropriate mechanisms for data/information/knowledge backup and recovery, in case of a system crash or malfunction.

5.3 CI as Complex Adaptive System

CI systems are complex by nature [70] and should be able to adapt to their environments, making such platforms complex adaptive systems (as suggested in S4 and S8). However, for a system to be complex adaptive, the system must exhibit adaptivity, self-organization, and emergence [60, 67, 70, 75].

Adaptivity means that the system or its components should allow constant changes over the period of its existence, depending upon the needs of its collective [70]. System developers should regularly update and evolve the platform by bringing in new technologies and services, based on user feedback and requirements, under the condition that these requirements are aligned with the system goals and objectives.

Self-organizing (A8) [47, 67] means the systems should be able to organize and re-organize its internal structure without the need of an external control [36, 72]. This behavior could be facilitated by allowing the creation of communities, where each member of the community would have a reputation that they could gain by providing useful contributions (in the form of insights, knowledge, or artifacts) and through up-votes/stars given to them by other members of their community. Such a reputation model can help create a structure within these communities, and therefore further interactions between such communities can lead to self-organizing behavior within the system.

Emergence (A9) in a system occurs when simple interactions among low-level system components give rise to new and unexpected patterns or properties, disparate from the properties of the system as a whole (based on the definition of emergence proposed by Damper [14]). In adaptive and self-organizing systems, regular modifications to the system and ever-changing user behavior may lead to the creation of unforeseen patterns, properties, or outcomes, thereby exhibiting emergent behavior.

6 COMPARATIVE CASE STUDIES

In this section, we evaluate the proposed generic model from Section 5.1 by examining six CI platforms with respect to the aforementioned model. The CI platforms were chosen on the basis of the following criteria: the platforms should belong to different disciplines/domains, the systems should be available for use (during the time of study), the platforms should have been published/discussed in scientific literature, the deliverables of the platforms should be available online, and lastly, the platforms should be recent or ongoing.

Based on these criteria, we identified six CI platforms (see Table 1 in the Supplementary Material). To analyze the platforms, we created user profiles on each of the platforms and observed system processes for create and decide activities; over the duration of 6 months, i.e., starting January 2018 to the end of June 2018. During this period, we interacted with the system as passive users. We created projects/ideas to analyze the creation process; however, we never submitted the projects/ideas for evaluation. We observed submissions for other participants and feedback from active users, and analyzed how the system communities and hierarchies work synchronously to come up with new contributions and innovative ideas. Additionally, we studied the available technical reports, scientific publications, FAQs, and other useful resources for each of the platforms. Aggregating our observations, we found that different aspects each of the six CI platforms could

Table 18. List of Studied CI Platforms and Their “What”

CI Platform	Year	What *	Domain
CAPSELLA	2016–2018	<i>IG:</i> Learn about new ICT technologies that can help improve agronomic practices. <i>CG:</i> Develop new ICT solutions, software and applications, and promote start-ups that can provide such solutions for agrifood business and farmers.	Agrobiodiversity
hackAIR	2016–2018	<i>IG:</i> Learn about the concentration of air pollutants (especially particulate matter) in cities and its effect on the health of local residents. <i>CG:</i> Provides citizens with real-time information about air pollution levels in their local vicinity and enables conversations for possible improvements in air quality.	Air pollution
openIDEO	Ongoing since 2010	<i>IG:</i> Demonstrate their skills and expertise to solve complex challenges, and learn from others’ work. <i>CG:</i> Tackle global challenges by developing innovative solutions using human-centric collaboration activities.	Innovation platform
Climate CoLab	Ongoing since 2009	<i>IG:</i> Participate in initiatives to help reach global climate goals. <i>CG:</i> Collaborate with other communities and experts, and help design/choose solutions to help identify sustainable growth initiatives.	Climate change
WikiCrimes	Ongoing since 2008	<i>IG:</i> Report criminal incidents. And keep track of crime rates in the local vicinity. <i>CG:</i> Assist governing bodies in validating reports of crimes provided by individuals. Help maintain a public record of all criminal activities.	Crime monitoring
Threadless	Ongoing since 2000	<i>IG:</i> Showcase their artistic ability by creating new T-shirt design. <i>CG:</i> Express community interest and select best T-shirt designs. Bring new and trending T-shirt designs to the marketplace.	Apparel design

* *IG:* Individual goals. *CG:* Community goals.

be described using our proposed generic model. Tables 18, 19, and 20 present the “What,” “Who,” and “Why-” “How” for each of the platforms, respectively.

6.1 What?

The goals of the six CI platforms can be summarized as follows:

The *CAPSELLA* project is designed to enable the creation of new ICT solutions for farmers and agricultural experts. The platform focuses on ICT contributions that facilitate the collection and exchange of data and experiences from individuals working in agriculture and bio-diversity.

Table 19. Comparative of CI Platforms—“Who”

CI Platform	Who			Open Data (Yes/No)
	Active (Crowd)	Active (Hierarchy)	Passive/Beneficiaries	
CAPSELLA	Farmers, food and seed communities	Agro-ecology, agri-food, ICT experts	Farmer communities, technology providers, other organizations	Yes
hackAIR	Citizens, open source communities	Environmental/health/educational organizations, scientific communities	Enterprises, local governments	Yes
OpenIDEO	Participants, innovators, alliances	Experts, challenge sponsors, advisory board	Participants (who only wish to participate in workshops)	No
Climate CoLab	Participants, community members	Fellows, judges	Government bodies, business organizations, civil society, individual citizens, consumers	No
WikiCrimes	Citizens	Agents, news media, government agencies	Citizens, government agencies	No
Threadless	Designers, consumers	Organization (Threadless)	Consumers	No

Table 20. Comparative of CI Platforms—“Why” and “How”

CI Platform	Why			How*		
	Intrinsic <i>Interest (I)/Passion (P)/ Knowledge (K)/Social cause (S)</i>	Extrinsic		Create		Decide (ID/GD)
		<i>Tangible</i>	<i>Intangible</i>	<i>Contest (CL/CB)</i>	<i>Voluntary (CL/CB)</i>	
CAPSELLA	IKS	Money	-	CL	Both	GD
hackAIR	IPKS	hackAIR sensors	Points, badges	CL	Both	GD
OpenIDEO	IPKS	Money	Glory	CL	Both	Both
Climate CoLab	IPKS	Money	Points	Both	CL	Both
WikiCrimes	IKS	-	Reputation	-	Both	Both
Threadless	IPK	Money	Design Quotient	CL	CL	Both

* CL: *Collection*. CB: *Collaboration*. ID: *Individual decision*. GD: *Group decision*.

hackAIR is designed as a platform where citizens can collect and access information about air quality in different parts of the world. The system empowers citizens by providing openly available DIY sensor designs, tool-kits and tutorials, thereby enabling citizens to be a part of the data collection process.

Similar to *hackAIR*, the *openIDEO* and *Climate CoLab* platforms deal with climate change and other environmental/societal challenges. However, both of these platforms are designed to enable the creation of new and innovative solutions by means of collaboration. While the contributions in *Climate CoLab* are focused toward global climate change goals, the contributions in *openIDEO* are focused more toward open innovation practices for societal change.

The *WikiCrimes* platform allows residents to anonymously report criminal activities in their local vicinity. This is especially useful in countries where citizens are not willing to contact the law enforcement agencies due to fear or lack of trust. The platform also allows its users to track the frequency and scale of criminal activities in different areas, thereby helping users in making better decisions when visiting specific locations.

Finally, the *Threadless* platform is meant for e-commerce and focuses on retail of apparels. The platform enables artists and designers to showcase their talent by sharing their T-shirt designs with the community. The best designs are then made available for sale on the Threadless marketplace, thereby providing artists and designers with a means of income.

We further elaborate the goals of these CI platforms as *individual* and *community goals* and domains in Table 18.

6.2 Who?

Table 19 presents the different actors of the analyzed CI platforms, segregated into three categories, namely, *active (crowd)*, *active (hierarchy)*, and *passive/beneficiaries* based on our proposed generic model. The table also indicates whether the platforms provide open data for future research or not.

6.3 Why and How?

Table 20 illustrates how each of the analyzed CI platforms motivates different kinds of actors using different sets of intrinsic and extrinsic motivators, and how different kinds of actors carry out different create and decide activities based on their roles within the system.

After mapping our observations (from each of the platforms) to our generic model, we found some interesting relationships between *actor* types, their *motivations*, and their *activities*:

- *Decide* activities are typically *intrinsically* motivated.
- *Contest (create)* activities by individuals of the *active (crowd)* are always *extrinsically* motivated. Whereas, *voluntary (create and decide)* contributions by individuals of the *active (crowd)* are always *intrinsically* motivated.
- *Voluntary (create)* contributions can be of two types: as data or information contributed by *crowd*, as in CAPSELLA, hackAIR, and WikiCrimes, or as feedback and suggestions given by *crowd* and members of *hierarchy* to help improve participants' contributions like in OpenIDEO, Climate CoLab, and Threadless.

7 THREATS TO VALIDITY

The primary threats to the validity of this Systematic Literature Review include bias in search strategy, bias in selection process, and inaccuracies in data extraction.

The selection of studies relied on the search strategy, which included the selection of search terms and literature resources, and the search process. The search terms were selected based on both the research questions and an initial literature review; followed by a three-step process to construct the search string as described in Section 1. We then chose four prominent academic databases of computer science and used the formulated search string to identify relevant literature. Table 2 presents the number and types of research articles identified from each of the academic databases. To avoid bias in our search strategy and to identify relevant technical reports, books, and theses, we conducted a manual search on Google Scholar.

To avoid bias in the study selection process, we first reviewed the titles and abstracts of the identified studies and then selected only those studies that fulfilled the inclusion criteria. We then studied these selected articles and manually checked their references to make sure that we did

not miss any relevant articles during the search process. Finally, the selected studies were then evaluated based on the quality assessment criteria. As a result of the study selection phase, we were able to identify the most relevant studies with respect to our research questions.

To eliminate inaccuracies in data extraction, each primary study was independently studied by all researchers and any disparities in findings were resolved through discussions. During the process, we found two pairs of studies, i.e., S1, S2 and S6, S9, which shared a couple of similarities. The first pair (S1, S2) described the characteristics of CI systems using similar classifications, while the second pair (S6, S9) was written by the same authors. By consensus, we decided to keep both pairs in our selected studies, as S1 and S2 described CI systems from different perspectives, whereas S6 and S9 provided different contributions.

8 CONCLUSION

The objective of this article was to analyze different collective intelligence models described in the scientific literature and to identify a generic model that could be utilized to design new CI platforms. To this end, we conducted a Systematic Literature Review, in which we identified 9,418 articles on collective intelligence models. Out of these articles, we selected 12 studies based on an exhaustive selection process. We then critically analyzed these selected studies and found that none of the models provided a generic view of CI systems, as each of the models was designed based on specific perspectives. And, the models that could potentially be used to design domain independent CI systems lacked granularity and needed to be researched further. So, to fill this research gap, we aggregated the components of the CI models described in the selected studies and proposed a unified framework for understanding CI systems. The proposed framework describes CI systems in three parts. First, a generic model, which describes CI systems as a combination of goals, staff, motivation, and processes, which are further described as types, interactions, and properties. Second, a list of requisites necessary for CI systems to work effectively. And third, guidelines that could enable complex adaptive behavior in CI platforms.

To evaluate if the proposed model could define CI systems from different domains, we selected a set of ongoing CI projects and observed user activities within the platform, over a duration of 6 months. After this, we systematically organized our observations and segregated them according to the different components of our proposed generic model. We found that our model successfully described the components of each of the CI platforms and revealed some interesting relations between the types of actors, their activities, and motivations. The evaluation of the proposed model also gave us the opportunity to present our unified CI framework by means of examples (i.e., six ongoing CI initiatives). It was imperative that we describe the components of these CI platforms in terms of the proposed CI model, so that both researchers and system designers/developers in the field could utilize our novel model to design and develop new CI systems. The 24 unique attributes that describe the proposed framework could provide initial insights to system designers and developers, and could be beneficial during the requirement elicitation process when developing new CI systems. We recognize that we need to further examine the proposed framework by comparing it to a larger set of CI platforms, as doing so would help us gain a deeper understanding about how the proposed framework could be used to design new CI systems. Additionally, we would like to evaluate the proposed framework by conducting qualitative interviews with domain experts and researchers working on upcoming CI initiatives. And finally, we would also like to investigate different trust and reputation models that could be utilized to reduce user bias within CI platforms, thereby enhancing user experience and enabling a smooth exchange of knowledge and information within communities.

REFERENCES

- [1] Lisa R. Anderson and Charles A. Holt. 1997. Information cascades in the laboratory. *The American Economic Review* 87, 5 (1997), 847–862. <http://www.jstor.org/stable/2951328>.
- [2] Kirsimarja Blomqvist. 1997. The many faces of trust. *Scandinavian Journal of Management* 13, 3 (1997), 271–286. DOI : [https://doi.org/10.1016/S0956-5221\(97\)84644-1](https://doi.org/10.1016/S0956-5221(97)84644-1)
- [3] Andre Boder. 2006. Collective intelligence: A keystone in knowledge management. *Journal of Knowledge Management* 10, 1 (2006), 81–93. DOI : <https://doi.org/10.1108/13673270610650120>
- [4] Eric Bonabeau. 2009. Decisions 2.0: The power of collective intelligence. *MIT Sloan Management Review* 50, 2 (Winter 2009), 45–52.
- [5] E. Bonabeau and C. Meyer. 2001. Swarm intelligence—A whole new way to think about business. *Harvard Business Review* 79, 5 (May 2001), 106+.
- [6] T. Bosse, C. M. Jonker, M. C. Schut, and J. Treur. 2006. Collective representational content for shared extended mind. *Cognitive Systems Research* 7, 2–3 (2006), 151–174. DOI : <https://doi.org/10.1016/j.cogsys.2005.11.007>
- [7] Daren C. Brabham. 2008. Crowdsourcing as a model for problem solving: An introduction and cases. *Convergence* 14, 1 (2008), 75–90. DOI : <https://doi.org/10.1177/1354856507084420>
- [8] Daren C. Brabham. 2010. Moving the crowd at Threadless. *Information, Communication & Society* 13, 8 (2010), 1122–1145. DOI : <https://doi.org/10.1080/13691181003624090>
- [9] David Budgen and Pearl Brereton. 2006. Performing systematic literature reviews in software engineering. In *Proceedings of the 28th International Conference on Software Engineering (ICSE'06)*. ACM, New York, NY, 1051–1052. DOI : <https://doi.org/10.1145/1134285.1134500>
- [10] Romina Cachia, Ramón Compañó, and Olivier Da Costa. 2007. Grasping the potential of online social networks for foresight. *Technological Forecasting and Social Change* 74, 8 (2007), 1179–1203. DOI : <https://doi.org/10.1016/j.techfore.2007.05.006>
- [11] Kyung Jin Cha, Yang Sok Kim, Byeonghwa Park, and Choong Kwon Lee. 2015. Knowledge management technologies for collaborative intelligence: A study of case company in Korea. *International Journal of Distributed Sensor Networks* 11, 9 (2015), 368273. DOI : <https://doi.org/10.1155/2015/368273>
- [12] Cary Cherniss. 2010. Emotional intelligence: Toward clarification of a concept. *Industrial and Organizational Psychology* 3, 2 (2010), 110–126. DOI : <https://doi.org/10.1111/j.1754-9434.2010.01231.x>
- [13] Prerna Chikersal, Maria Tomprou, Young Ji Kim, Anita Williams Woolley, and Laura Dabbish. 2017. Deep structures of collaboration: Physiological correlates of collective intelligence and group satisfaction. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW'17)*. ACM, New York, NY, 873–888. DOI : <https://doi.org/10.1145/2998181.2998250>
- [14] R. I. Dampier. 2000. Editorial for the special issue on ‘Emergent Properties of Complex Systems’: Emergence and levels of abstraction. *International Journal of Systems Science* 31, 7 (2000), 811–818. DOI : <https://doi.org/10.1080/002077200406543>
- [15] Miguel de Castro Neto and Ana Espirto Santo. 2012. Emerging collective intelligence business models. In *MCIS 2012 Proceedings*. Mediterranean Conference on Information Systems. <https://aisel.aisnet.org/mcis2012/14>.
- [16] Tony Diggle. 2013. Water: How collective intelligence initiatives can address this challenge. *Foresight* 15, 5 (2013), 342–353. DOI : <https://doi.org/10.1108/FS-05-2012-0032>
- [17] Anhai Doan, Raghu Ramakrishnan, and Alon Y. Halevy. 2011. Crowdsourcing systems on the World-Wide Web. *Communications of the ACM* 54, 4 (2011), 86. DOI : <https://doi.org/10.1145/1924421.1924442>
- [18] Ketry Gorete Farias dos Passos and Edna Lúcia da Silva. 2012. Effect of collective intelligence in organizations. *Transinformação* 24, 2 (May 2012), 127–136. DOI : <https://doi.org/10.1590/s0103-37862012000200005>
- [19] Vasco Furtado, Leonardo Ayres, Marcos de Oliveira, Eurico Vasconcelos, Carlos Caminha, Johnatas D’Orleans, and Mairon Belchior. 2010. Collective intelligence in law enforcement—The WikiCrimes system. *Information Sciences* 180, 1 (2010), 4–17. DOI : <https://doi.org/10.1016/j.ins.2009.08.004>
- [20] Jing Gao, Qi Li, Bo Zhao, Wei Fan, and Jiawei Han. 2015. Truth discovery and crowdsourcing aggregation: A unified perspective. *PVLDB* 8, 12 (2015), 2048–2049. DOI : <https://doi.org/10.14778/2824032.2824136>
- [21] Sandro Georgi and Reinhard Jung. 2012. Collective intelligence model: How to describe collective intelligence. In *Advances in Intelligent and Soft Computing*. Vol. 113. Springer, 53–64. DOI : https://doi.org/10.1007/978-3-642-25321-8_5
- [22] Jim Giles. 2005. Internet encyclopaedias go head to head. *Nature* 438, 7070 (Dec. 2005), 900–901. DOI : <https://doi.org/10.1038/438900a>
- [23] Jerome C. Glenn. 2013. Collective intelligence and an application by the millennium project. *World Futures Review* 5, 3 (2013), 235–243. DOI : <https://doi.org/10.1177/1946756713497331>
- [24] Vijay Govindarajan and Chris Trimble. 2011. The CEO’s role in business model reinvention. *Harvard Business Review* 89, 1–2 (2011), 108–14.

- [25] Antonietta Grasso and Gregorio Convertino. 2012. Collective intelligence in organizations: Tools and studies. *Computer Supported Cooperative Work (CSCW)* 21, 4 (01 Oct 2012), 357–369. DOI: <https://doi.org/10.1007/s10606-012-9165-3>
- [26] Dawn Gregg. 2009. Developing a collective intelligence application for special education. *Decision Support Systems* 47, 4 (2009), 455–465. DOI: <https://doi.org/10.1016/j.dss.2009.04.012>
- [27] Dawn G. Gregg. 2010. Designing for collective intelligence. *Communications of the ACM* 53, 4 (April 2010), 134–138. DOI: <https://doi.org/10.1145/1721654.1721691>
- [28] Tom Gruber. 2008. Collective knowledge systems: Where the social web meets the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web* 6, 1 (2008), 4–13. DOI: <https://doi.org/10.1016/j.websem.2007.11.011>
- [29] Friedrich August Hayek. 1945. The use of knowledge in society. *The American Economic Review* 35, 4 (1945), 519–530.
- [30] Sylvia Ann Hewlett, Melinda Marshall, and Laura Sherbin. 2013. How diversity can drive innovation. *Harvard Business Review* 91, 12 (2013), 30–30.
- [31] Lu Hong and Scott E. Page. 2004. Groups of diverse problem solvers can outperform groups of high-ability problem solvers. *Proceedings of the National Academy of Sciences* 101, 46 (2004), 16385–16389. DOI: <https://doi.org/10.1073/pnas.0403723101>
- [32] Luca Iandoli. 2009. Leveraging the power of collective intelligence through IT-enabled global collaboration. *Journal of Global Information Technology Management* 12, 3 (2009), 1–6. DOI: <https://doi.org/10.1080/1097198X.2009.10856494>
- [33] Luca Iandoli, Mark Klein, and Giuseppe Zollo. 2009. Enabling on-line deliberation and collective decision-making through large-scale argumentation. *International Journal of Decision Support System Technology* 1, 1 (Jan. 2009), 69–92. DOI: <https://doi.org/10.4018/jdsst.2009010105>
- [34] Joshua Introne, Robert Laubacher, Gary Olson, and Thomas Malone. 2011. The climate CoLab: Large scale model-based collaborative planning. In *Proceedings of the 2011 International Conference on Collaboration Technologies and Systems (CTS'11)*. 40–47. DOI: <https://doi.org/10.1109/CTS.2011.5928663>
- [35] Caroline Just. 2011. A review of literature on the general factor of personality. *Personality and Individual Differences* 50, 6 (2011), 765–771. DOI: <https://doi.org/10.1016/j.paid.2011.01.008>
- [36] Stuart A. Kauffman. 1993. *The Origins of Order: Self-organization and Selection in Evolution*. Oxford University Press.
- [37] Barbara A. Kitchenham and S. Charters. 2007. *Guidelines for Performing Systematic Literature Reviews in Software Engineering*. Technical Report EBSE-2007-01. Keele University.
- [38] Gary A Klein. 2004. *The Power of Intuition: How to Use Your Gut Feelings to Make Better Decisions at Work*. Crown Business.
- [39] A. Kornrumpf and U. Baumöl. 2014. A design science approach to collective intelligence systems. In *2014 47th Hawaii International Conference on System Sciences*. 361–370. DOI: <https://doi.org/10.1109/HICSS.2014.53>
- [40] Evangelos Kosmidis et al. 2018. hackAIR: Towards raising awareness about air quality in Europe by developing a collective online platform. *ISPRS International Journal of Geo-Information* 7, 5 (2018). DOI: <https://doi.org/10.3390/ijgi7050187>
- [41] Hélène Landemore and Jon Elster. 2012. *Collective Wisdom: Principles and Mechanisms*. Cambridge University Press. DOI: <https://doi.org/10.1017/CBO9780511846427>
- [42] Vita Lanfranchi and Neil Ireson. 2009. User requirements for a collective intelligence emergency response system. In *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology (BCS-HCI'09)*. British Computer Society, Swinton, UK, 198–203. <http://dl.acm.org/citation.cfm?id=1671011.1671035>.
- [43] Pierre Levy. 1997. *Collective Intelligence: Mankind's Emerging World in Cyberspace*. Perseus Books, Cambridge, MA.
- [44] Pierre Lévy. 2010. From social computing to reflexive collective intelligence: The IEML research program. *Information Sciences* 180, 1 (2010), 71–94. DOI: <https://doi.org/10.1016/j.ins.2009.08.001>
- [45] Randall M. Livingstone. 2016. Models for understanding collective intelligence on wikipedia. *Social Science Computer Review* 34, 4 (Aug. 2016), 497–508. DOI: <https://doi.org/10.1177/0894439315591136>
- [46] Jan Lorenz, Heiko Rauhut, Frank Schweitzer, and Dirk Helbing. 2011. How social influence can undermine the wisdom of crowd effect. *Proceedings of the National Academy of Sciences* 108, 22 (2011), 9020–9025. DOI: <https://doi.org/10.1073/pnas.1008636108>
- [47] Shuangling Luo, Haoxiang Xia, Taketoshi Yoshida, and Zhongtuo Wang. 2009. Toward collective intelligence of online communities: A primitive conceptual model. *Journal of Systems Science and Systems Engineering* 18, 2 (01 June 2009), 203–221. DOI: <https://doi.org/10.1007/s11518-009-5095-0>
- [48] Ioanna Lykourantzou, Dimitrios J. Vergados, Epaminondas Kapetanios, and Vassili Loumos. 2011. Collective intelligence systems: Classification and modeling. *Journal of Emerging Technologies in Web Intelligence* 3, 3 (Aug. 2011). DOI: <https://doi.org/10.4304/jetwi.3.3.217-226>
- [49] Ioanna Lykourantzou, Dimitrios J. Vergados, and Vassili Loumos. 2009. Collective intelligence system engineering. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems (MEDES'09)*. ACM, New York, NY, Article 20, 7 pages. DOI: <https://doi.org/10.1145/1643823.1643848>

- [50] T. W. Malone and M. S. Bernstein. 2015. *Handbook of Collective Intelligence*. MIT Press. 2015029234 <https://books.google.ee/books?id=Px3iCgAAQBAJ>.
- [51] Thomas W. Malone and Mark Klein. 2007. Harnessing collective intelligence to address global climate change. *Innovations: Technology, Governance, Globalization* 2, 3 (2007), 15–26. DOI : <https://doi.org/10.1162/itgg.2007.2.3.15>
- [52] Thomas W. Malone, Robert Laubacher, and Chrysanthos N. Dellarocas. 2009. Harnessing crowds: Mapping the genome of collective intelligence. *SSRN Electronic Journal* (2009). DOI : <https://doi.org/10.2139/ssrn.1381502>
- [53] Kurt Matzler, Andreas Strobl, and Franz Bailom. 2016. Leadership and the wisdom of crowds: How to tap into the collective intelligence of an organization. *Strategy & Leadership* 44, 1 (2016), 30–35. DOI : <https://doi.org/10.1108/SL-06-2015-0049> arXiv:<https://doi.org/10.1108/SL-06-2015-0049>
- [54] Gavin McCrory, Carina Veeckman, and Laurence Claeys. 2017. Citizen science is in the air—Engagement mechanisms from technology-mediated citizen science projects addressing air pollution. In *Internet Science*. Springer International Publishing, Cham, 28–38. DOI : https://doi.org/10.1007/978-3-319-70284-1_3
- [55] Lev Muchnik, Sinan Aral, and Sean J. Taylor. 2013. Social influence bias: A randomized experiment. *Science* 341, 6146 (2013), 647–651. DOI : <https://doi.org/10.1126/science.1240466>
- [56] J. Musil, A. Musil, D. Weyns, and S. Biffl. 2015. An architecture framework for collective intelligence systems. In *2015 12th Working IEEE/IFIP Conference on Software Architecture (WICSA'15)*. 21–30. DOI : <https://doi.org/10.1109/WICSA.2015.30>
- [57] Kimberly Nehls. 2015. Crowdsourcing. *Learning, Media and Technology* 40, 1 (2015), 123–126. DOI : <https://doi.org/10.1080/17439884.2014.929144>
- [58] Van Du Nguyen and Ngoc Thanh Nguyen. 2018. Intelligent collectives: Theory, applications, and research challenges. *Cybernetics and Systems* 49, 5-6 (2018), 261–279. DOI : <https://doi.org/10.1080/01969722.2017.1418254>
- [59] Ikujiro Nonaka and Takeuchi Hirotaka. 1995. *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press. 94040408 <https://books.google.ee/books?id=B-qxrPaU1-MC>.
- [60] Julio M. Ottino. 2004. Engineering complex systems. *Nature* 427, 6973 (2004), 399. DOI : <https://doi.org/10.1038/427399a>
- [61] S. E. Page and Princeton University Press. 2007. *The Difference: How the Power of Diversity Creates Better Groups, Firms, Schools, and Societies*. Princeton University Press. 2006044678 <https://books.google.ee/books?id=FAFVHnJ7uK0C>.
- [62] Chloe Parker. 2015. The IWA WaterWiki: An open-access publishing platform providing free content on all aspects of water, wastewater and the environment. *Environment and Urbanization* 27, 1 (2015), 137–138. DOI : <https://doi.org/10.1177/0956247814547172>
- [63] Michael A. Peters and Richard Heraud. 2015. Toward a political theory of social innovation: Collective intelligence and the co-creation of social goods. 3, 3 (2015), 7–23. <https://researchcommons.waikato.ac.nz/handle/10289/9569>.
- [64] Rolf Pfeifer, Jan Henrik Sieg, Thierry Bücheler, and Rudolf Marcel Fuchsli. 2010. Crowdsourcing, open innovation and collective intelligence in the scientific method: A research agenda and operational framework. (2010). DOI : <https://doi.org/10.21256/zhaw-4094>
- [65] P. Clint Rogers, Stephen W. Liddle, Peter Chan, Aaron Doxey, and Brady Isom. 2007. WEB 2.0 learning platform: Harnessing collective intelligence. *Turkish Online Journal of Distance Education* 8, 3 (2007), 16–33. http://tojde.anadolu.edu.tr/makale_goster.php?id=348.
- [66] Frantz Rowe. 2014. What literature review is not: Diversity, boundaries and recommendations. *European Journal of Information Systems* 23, 3 (01 May 2014), 241–255. DOI : <https://doi.org/10.1057/ejis.2014.7>
- [67] Juho Salminen. 2015. *The Role of Collective Intelligence in Crowdsourcing Innovation*. PhD dissertation. Lappeenranta University of Technology.
- [68] H. Santos, L. Ayres, C. Caminha, and V. Furtado. 2012. Open government and citizen participation in law enforcement via crowd mapping. *IEEE Intelligent Systems* 27 (2012), 63–69. DOI : <https://doi.org/10.1109/MIS.2012.80>
- [69] Detlef Schoder, Peter A. Gloor, and Panagiotis Takis Metaxas. 2013. Social media and collective intelligence—Ongoing and future research streams. *KI - Künstliche Intelligenz* 27, 1 (1 Feb. 2013), 9–15. DOI : <https://doi.org/10.1007/s13218-012-0228-x>
- [70] Martijn C. Schut. 2010. On model design for simulation of collective intelligence. *Information Sciences* 180, 1 (2010), 132–155. DOI : <https://doi.org/10.1016/j.ins.2009.08.006> Special Issue on Collective Intelligence.
- [71] Toby Segaran. 2007. *Programming Collective Intelligence : Building Smart Web 2.0 Applications*. O'Reilly, Beijing/Sebastapol, CA.
- [72] V. Singh, G. Singh, and S. Pande. 2013. Emergence, self-organization and collective intelligence—Modeling the dynamics of complex collectives in social and organizational settings. In *2013 UKSim 15th International Conference on Computer Modelling and Simulation*. 182–189. DOI : <https://doi.org/10.1109/UKSim.2013.77>
- [73] Aelita Skarzauskiene and Monika Maciuliene. 2015. Modelling the index of collective intelligence in online community projects. In *International Conference on Cyber Warfare and Security*. Academic Conferences International Limited, 313.

- [74] Daniel Stieger, Kurt Matzler, Sayan Chatterjee, and Florian Ladstaetter-Fussenegger. 2012. Democratizing strategy: How crowdsourcing can be used for strategy dialogues. *California Management Review* 54, 4 (2012), 44–68. DOI: <https://doi.org/10.1525/cm.2012.54.4.44>
- [75] Tim Sullivan. 2014. Embracing Complexity. Retrieved on March 8, 2019 from <https://hbr.org/2011/09/embracing-complexity>.
- [76] Shweta Suran, V. Pattanaik, Sadok B. Yahia, and Dirk Draheim. 2019. Exploratory analysis of collective intelligence projects developed within the EU-Horizon 2020 framework. In *Computational Collective Intelligence*. Springer International Publishing, 285–296.
- [77] James Surowiecki. 2005. *The Wisdom of Crowds*. Anchor.
- [78] Guy Theraulaz and Eric Bonbeau. 1999. A brief history of stigmergy. *Artificial Life* 5, 2 (April 1999), 97–116. DOI: <https://doi.org/10.1162/106454699568700>
- [79] O'Reilly Tim. 2007. *What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software*. MPRAPaper. University Library of Munich, Germany. <https://EconPapers.repec.org/RePEc:pra:mprapa:4578>.
- [80] John P. van Gigch. 1991. *System Design Modeling and Metamodeling*. Springer US. DOI: <https://doi.org/10.1007/978-1-4899-0676-2>
- [81] Dimitrios J. Vergados, Ioanna Lykourantzou, and Epaminondas Kapetanios. 2010. A resource allocation framework for collective intelligence system engineering. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems (MEDES'10)*. ACM, New York, NY, 182–188. DOI: <https://doi.org/10.1145/1936254.1936285>
- [82] Jennifer H. Watkins. 2007. Prediction markets as an aggregation mechanism for collective intelligence. Retrieved from <https://escholarship.org/uc/item/8mg0p0zc>.
- [83] Sean Wise, Robert A. Paton, and Thomas Gegenhuber. 2012. Value co-creation through collective intelligence in the public sector: A review of US and European initiatives. *VINE* 42, 2 (2012), 251–276. DOI: <https://doi.org/10.1108/03055721211227273>
- [84] Michael A. Woodley and Edward Bell. 2011. Is collective intelligence (mostly) the General Factor of Personality? A comment on Woolley, Chabris, Pentland, Hashmi and Malone (2010). *Intelligence* 39, 2 (2011), 79–81. DOI: <https://doi.org/10.1016/j.intell.2011.01.004>
- [85] Anita Williams Woolley, Christopher F. Chabris, Alex Pentland, Nada Hashmi, and Thomas W. Malone. 2010. Evidence for a collective intelligence factor in the performance of human groups. *Science* 330, 6004 (2010), 686–688. DOI: <https://doi.org/10.1126/science.1193147>
- [86] Koji Zettsu and Yasushi Kiyoki. 2006. Towards knowledge management based on harnessing collective intelligence on the web. In *Managing Knowledge in a World of Networks*. Springer, Berlin, 350–357. DOI: https://doi.org/10.1007/11891451_31

Received March 2019; revised October 2019; accepted October 2019

Appendix 6

VI

S. A. Peious, S. Suran, V. Pattanaik, and D. Draheim. Enabling sensemaking and trust in communities: An organizational perspective. In *Proceedings of the 23rd International Conference on Information Integration and Web-Based Applications & Services, iiWAS '21*, page 1–9, New York, NY, USA, 2021. Association for Computing Machinery

Enabling Sensemaking and Trust in Communities: An Organizational Perspective

Sijo Arakkal Peious
Information Systems Group,
Tallinn University of Technology
Tallinn, Estonia
sijo.arakkal@taltech.ee

Vishwajeet Pattanaik
Information Systems Group,
Tallinn University of Technology
Tallinn, Estonia
vishwajeet.pattanaik@taltech.ee

Shweta Suran
Information Systems Group,
Tallinn University of Technology
Tallinn, Estonia
shweta@taltech.ee

Dirk Draheim
Information Systems Group,
Tallinn University of Technology
Tallinn, Estonia
dirk.draheim@taltech.ee

ABSTRACT

The large volume of information being produced in organizations today poses new challenges to the accuracy and effectiveness of any organizations' decision-making processes. These challenges, namely sensemaking and trust, can critically impact the decision-making processes, even if the organizations are relying on business intelligence (BI) strategies. Given the critical impact an organizations' BI can have on its sustainability and thus its success, in this work, we attempt to draw insights from the literature on collective intelligence and, based on these, present a novel artifact that aims to empower organizations' BI by supporting the organizations' employees in establishing trust and sense when working up with new ideas and solutions. The proposed artifact utilizes a novel reputation model, which calculates reputation based on an individual's area of expertise and reputation score, in order to assist in establishing trust among system users, and thus helps improve decision-making processes.

CCS CONCEPTS

• **Information systems** → **Crowdsourcing**; • **Software and its engineering** → *Development frameworks and environments*; *Use cases*; *Abstraction, modeling and modularity*; *Designing software*.

KEYWORDS

business intelligence, collective intelligence, crowdsourcing, sensemaking, trust, reputation model

ACM Reference Format:

Sijo Arakkal Peious, Shweta Suran, Vishwajeet Pattanaik, and Dirk Draheim. 2021. Enabling Sensemaking and Trust in Communities: An Organizational Perspective. In *The 23rd International Conference on Information Integration*

and Web Intelligence (iiWAS2021), November 29-December 1, 2021, Linz, Austria. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3487664.3487678>

1 INTRODUCTION

Today's business organizations face endless instabilities and volatilities, which can lead to creation of massive volumes of data; being produced by organizations both internally and externally [!REF]. To harness the possibilities of this transformation, several organizations now aspire (but often even struggle) to convert these large volumes of existing data into a clear understandable chunks that could be utilized in their business processes. In order to achieve this businesses reply on Business Intelligence (BI); a strategy that enables organizations to examine their past actions and decisions, and thus consequently, predict the future. BI denotes a wide range of technologies, processes and applications that assist organizations in gathering, storing, evaluating, and granting access to data for refining business's processes and over-all decision-making [17, 38]. It aids organizations by continuously collecting and analyzing organizational information (including performance metrics) and assists by making the decision-making processes more efficient.

Although BI is a powerful tool and can be typically used in an organization's almost all decision-making processes (both long-term and short-term), however, business organizations today only use BI for day-to-day (i.e., short-term) decision-making [20] and, presently, BI abilities are not necessarily utilized for identifying the organizations' long-term progression, which could indeed help them in improving their methods when undertaking tactical decisions [8]. Another problem that can arise when using BI (which is also often discussed in literature) is sensemaking [35]; this is a key precondition to reach an informed decision and is based on the prior actions of humans [3]. This is to say, that given BI relies on both machine intelligence and human intelligence, when assisting organizations in decision-making; the humans involved in analysis tasks can often get confused by the lack of sense in an idea or an outcome.

Now given that by gaining a better 'sense' of the organization overall, managers (and other decision makers) could better understand their business's organizational environment and hence make healthier decisions [34]; BI applications and related strategies can

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

iiWAS2021, November 29-December 1, 2021, Linz, Austria

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-9556-4/21/11...\$15.00

<https://doi.org/10.1145/3487664.3487678>

play a critical part in sensible decision making, and even added advantages beyond conventional decision-making. It is key to note here that, the decisions that are made using BI should be both sensible and explainable and should cover various potential possibilities.

In BI, data/information is used to create reports, summarize past actions, forecast actions, and to understand current and future risks. When relying on BI, the precision of predictions (made using BI strategies) depends on the quality of the information and its sources [39]; if the information and its source are not trustworthy, the entire action and its outputs can become futile (even counter-productive). Managers and decision makers who use these outputs typically understand on-going scenarios, and hence create productive decisions or implement their decisions keeping the scenarios in mind [10]; however, a key factor that can influence the decision-making process in such scenarios is ‘trust’; specially since humans are involved in the process. Consider this for example, if managers from distinct departments/sections of an organization are working together on creating a solution for given scenario, individuals who have encountered similar scenarios before might be able to contribute more to the solution, however, if the managers are not aware of the past experiences of their colleagues, they might end up not considering ideas of the individual that could contribute the most. This is in line with literature, where researchers have found that, when working together in groups humans tend to make better decisions when there is trust among group-members [29].

That said, in this work we attempt to tackle the above mentioned issues of sensemaking and trust, and propose a novel platform designed as discussion forum oriented towards managers, decision-makers and other employees working in organizations. To achieve this, we draw influence from another domain (one that dates back to Aristotle), that is, Collective Intelligence (CI, defined as, “groups of individuals acting collectively in ways that seem intelligent” [21]); as the domain has recently gained traction in a wide variety of domains [29]. So much so, that it is actively being used by both governing bodies and organizations today; not only to collect citizen/end-user feedback, but also in the design processes for solving critical issues and developing new products, respectively (for example, in Crowd4Roads and CAPSELLA [30], and openIDEO and Threadless [29]). In general, through its fundamental concepts of collection and collaboration, CI has allowed organizations to make better use of the (collective) intelligence of their employees and their users, and thus helps enhance their decision-making processes, when gathering information from numerous sources and creating valuable outputs using CI methods.

With this in mind, the overall aim of this study, is to discover how BI strategies could contribute to better decision-making in presence of sensemaking and trust. The study mainly focuses on the organizations’ employee’s perspective and tries to identify factors that generate trust between employees and attempts to understand how this trust helps in sensible decision-making processes. In particular, we would like to answer to answer the following research questions:

Q1: *Can we solve the issues of trust and sensemaking in BI using the concepts from CI?*

Q2: *How can we design a reputation model for such a BI system while solving well-known challenges related to reputation in CI platforms?*

The remaining paper is organized as follows, in Section 2, background and related work of BI, CI, trust and reputation systems are described. Then, Section delves into the novel reputation model of trust and sensemaking, and Section discusses the proposed artifact (i.e., the CI-Forum). In Section 5 we describe the evaluation process for the developed forum and reputation model; and finally, Section provides a brief discussion on the findings of this work before concluding the paper.

2 BACKGROUND AND RELATED WORK

Business organizations’ performance relies on real-time and effective organizational information. BI systems analyze this information and identify shortcomings and problems within an organization, they provide businesses with insights and suggestions in real-time and support decision-makers in coming up with better conclusions; which subsequently helps organizations sustain and improve productivity [2, 33]. By implementing innovative ideas and new technologies in their processes, businesses can achieve competitive advantage and success in rapidly changing business conditions [13, 22].

2.1 Business Intelligence

The decision-making processes change according to the information businesses are using to make decisions within their organizations [18]. We can characterize a BI system as a framework that collects, makes modification and generates business’s organizational information from different resources. This reduces the time required for analyzing important business information and helps managers to make efficient decisions that can be utilized to improve business strategies. BI is the process of combining different series of actions and business information to provide a competitive advantage to business organizations by helping decision-makers [25]. It is a system and generates answers to support decision-makers to understand the economic situations of the business organizations [23]. Conventionally, BI uses methodological models and numerical functionalities for analysis, used for mining valuable business information and data from basic information to help managers and decision-makers [31]. These business information mining processes and analysis procedures enhance forecasting and help decision-makers understand the progression and problems of any business organization [26].

2.2 Collective Intelligence and Crowdsourcing

General intelligence, as understood by psychologists is the (single) statistical factor that predicts variance in performance, when an individual performs some cognitive tasks (e.g., [11]); it includes an individuals capacity for logic, understanding, learning, reasoning, planning, creativity, critical thinking, problem-solving and many other aspects. When a group of individuals (human or machine) work together and use their individual intelligence, the aggregated intelligence of the group can be understood as CI.

In Information and Communications Technologies (ICT), CI has several definitions (for example, the most prominent ones are by Levy [11] and Malone [21]); each defines CI as having, three components: “individuals (with data/information/knowledge), coordination and collaboration activities (according to a predefined set of rules), and means/platform for real-time communication (viz.,

hardware/software)—together these “enable intelligent behavior in groups or crowds” [29]. That said, that advent of the Internet has allowed for mobilization and harnessing of CI in truly novel ways, and this has enabled creation of web-based group discussion platforms that play a key role decision making today [28]. This has opened the gates to a wide variety of emerging research topics, including for example, research where scientists and academicians are trying to understand the influence of group discussion platforms on performance improvement in the quality, efficiency, and effectiveness of decision-making when using such platforms [24]. Some researchers are also focusing on how users behave, group members carry out activities, and knowledge that is generated on group discussion platform by a user and their groups. There also have been studies which focuses on collaborative IT solutions and group discussion systems (designed as web-based platforms), and aim to explain how BI is being used in business organizational context [8].

Another application of CI, that is gaining tremendous interest in research is crowdsourcing (defined as, process where a group of people work together and carry out a task, typically involving collection of data/information or building a solution; that was conventionally done by a single individual [7]. CI (also, crowdsourcing) involves group of people working together, but its key that the individual members of group are diverse [29]. Some researchers have expressed that the main aim of crowdsourcing is to distribute the task of one person to a group of people, and by doing this the overall workload can be divided and hence the task can be carried out effortlessly [6]. Such crowdsourcing activities are divided into three categories. First, directed crowdsourcing, where, a coordinator asks a specific question (with relevant explanation) oriented towards participants, and participants earn some kind of rewards or benefits to the effort and time they contribute. The second category is self-directed, where, participants contribute due to intrinsic motivations. Here participants comes to a common platform and discuss various topics according to their volition and try to come up with decisions or actions according based on the topic at hand. The third and final category is passive, where crowdsourcing is only a side effect of output produced by some action. Here, participant are not obliged to generate the output, or might not be even aware that are participating in a crowdsourced system [36].

A first popular example of crowdsourcing that has is often discussed in literature is the Goldcorp Inc.’s initiative from the year 2000, where they used crowdsourcing to identify gold mines in the Red Lake. The participants were awarded around 0.5 million, and Goldcorp agreed to share the information about the gold mines if they were able to find 6 million ounces of gold, from an identified site. Geologists and engineers from various counties started analysing the information provided by Goldcorp and the company started to receive replies (i.e., potential sites with gold) in a short amount of time. The results produced by participants were verified by a panel decided by Goldcorp, and the end of the competition the panel members were surprised by the both the creativity of the participants and the results produced by them. Goldcorp drilled at the best 5 locations suggested by participants, and found gold from at each of the locations. A key finding of the competition was that participants were able to find gold from all of these locations, without even the locations once. The competition also illustrated

how intelligent individuals are, and that by utilizing humans (collective) intelligence combined with technology, organizations could come up with novel and innovative solutions (which could not be achieved conventionally) [5, 37].

2.3 Trust

Reputation and trust are considered key factors of a civilized society [12]. In CI systems too, trust is considered a key property [14, 29]. The success rate of a CI platform can be judged by measuring the trust and openness among the users [4, 14]. An easy method to assess the trustworthiness can be to just ask the users if they trust the source of information [32]. Dworken et al. [15] explained how trust perceived by organizations using examples from the news industry. They claimed that news coverage over the years has changed dramatically, and this is because users have started to analyze both the news and its source to check the reliability of the information [15]. Trust is also a key component in decision making as well as in collaborative working environments [12]. Trust is the belief that the trusted person or the organization will accomplish a particular task according to the task givers expectation [16]. BI applications provide trustable descriptions of various business situations and deliver numerous outcomes for understanding business organizational risks; whoever, as we eluded to earlier, even with the trustable nature of BI applications, trust and sensemaking still remain a challenge to some extent.

2.4 Reputation Systems

Reputation systems are mathematical functions used to calculate a user or objects trustworthiness or value as perceived by fellow users, and is calculated based on user feedback (which can represented using up-votes, stars, like etc.). Theses user score provided by fellow users can be used as a benchmark to identify the level of user trustworthy, and the aggregate of votes and feedback are considered as the reputation score. Literature indicates that theses votes/feedback and thus reputation score can often be violated, thus providing untruthful feedback to gain reputation (supporting non-worthy users) or to decrease the reputation of other users [27]. Reputation systems also face numerous other challenges [1], for instance, Sybil attacks, where attackers (or malicious users) create multiple fake accounts to up-vote their contributions in order to gain higher reputation score, or excessive use of self-promotion, or users with high negative reputations tend to delete old accounts and create new ones (this is referred to as whitewashing). A solution to whitewashing however is that the time duration it takes for an individual to gain reputation can be studied (as was done in [19]), as true good reputation typically only grows gradually. Another challenge to reputation systems is oscillation attack, where, the attacker creates a user account and behaves fairly to achieve good reputation, and then changes their behaviour, hence misleading noble users who trusting the reputation of the attacker [9]. This these challenges in mind, in this work, we aim to develop a novel reputation model that would attempt to tackle some of the challenges described above.

To summarize, this section presented a brief background of literature of BI, CI and reputation systems; this is critical as the review of the literature allowed us to illustrate the purpose of the study, the questions, limits and advantages. It also provides theoretical

viewpoints, current views for identifying the study questions and a review of related experimental studies concerning the respective fields. The following section explains the proposed reputation model and how it is different from existing models and systems.

3 NOVEL REPUTATION MODEL

The study proposes a novel approach to the reputation system which aims to avoid the problems explained in the reputation system's literature review. The proposed approach follows a decentralized reputation system. To some extent, this model is similar to existing distributed reputation models like the one used by 'Stack Overflow'. The users give feedback through positive and negative votes (i.e., up-down votes). Whenever a user receives a vote, their reputation score is altered dynamically according to the received votes. In the proposed method, users would not get the same score every time they receive an up/down vote; instead the amount of score that would be added or reduced would depend on the reputation score of the user giving the vote. This means, if a user has a high reputation and they give an up-vote to another user then the receiving user's reputation score would increase by a higher value, and if the user giving the up-vote does not have any reputation then the receiving user will get the minimal increase in their reputation score. In this approach, the overall score is not calculated while making the vote; but rather the votes are calculated with respect to the category tags (on the individuals profile, i.e., only the topics the user is familiar with) and points are also calculated according to these category tags.

In the proposed reputation model scores are calculated based on the category tags. Whenever a user casts their vote, the system first checks for the reputation score of that user according to the category. If the user has a reputation score, then the system divides that score (that will be added to the receiver's reputation) using the total number of votes that the user has received for the particular category. If the calculated score is less than the minimum score, then the receiving user receives the minimum score else they receive the calculated score.

Take for example the following scenario, let us assume there are ten users:

$A_1, A_2, A_3, \dots, A_{10}$, the minimum reputation score is 1 and we have three categories C_0, C_1 , and C_2 . At the starting time, T_0 everyone's reputation score is 1. If at a certain time T_1 , 5 users are making positive votes for A_6 in respect to category C_0 , then his or her reputation score will be $1 + (1 \times 5) = 6$. This score is the overall and C_0 's reputation score. At time T_2 , if A_6 is casting a positive vote for A_7 with respect to category C_0 , then A_7 will receive $1 + (6/5) = 2.2$, 1 is a minimum score of A_7 , 6 is reputation score and 5 is the total number of votes for A_6 with respect to C_0 . At time T_3 , if A_6 is casting a positive vote for A_8 with respect to category C_1 , then A_8 will get $1 + 1 = 2$. At time T_4 , if A_6 is getting a positive vote from A_9 with respect to C_2 , then $1 + 1 = 2$ is added to both category C_2 and the overall reputation score. At the time of T_4 , the reputation scores of A_6 are, the overall reputation score is 8, category C_0 is 6, category C_1 is 1 and C_2 is 2.

When generating scores for negative votes (i.e., down votes) the exact same strategy is used, but with subtraction used instead of addition.

To summarise, in this section, a novel reputation model has been described. The main advantage of the proposed reputation system is, that users can identify the expertise of every user by viewing an overall reputation score and separate score based on every category (the individual contributes/has contributed to). Now to validate this reputation model we have created an artifact, which we delve into in Section 4.

4 PROPOSED CI-FORUM

To study how sensemaking and trust can influence on user behaviour, and to evaluate the previously proposed reputation model here we present as discussion forum (named "CI-Forum"). The proposed artifact allows users to post questions and reply to the questions posted by other users. Users can the platform share knowledge and help other users to solve problems. Users can up-vote or down-vote other users comments and feedback, which in turn is used to calculate user reputation. Users can view posts by using filters, for example sorted based on the reputation scores of the user who posted the question/comment; and thus should be able to identify individuals experts (based on the best answers/comments). The primary notion behind the artifact is that such a CI based forum could potentially be used in line with BI strategies, and would allow organizations to use the collective intelligence of their employees when carrying out decision-making processes.

4.1 Coding and Implementation

The user interface for the artifact is designed using HTML, CSS and JavaScript. To send and receive data, AJAX POST method is used. The CI-Forum website communicates with the server and collects information in the form of JSON objects and files. To make the design process easier and to master coding, pages are used. On the server side, C# is used as the main programming language, together with a layered architecture. The application consists of four layers, i.e., a main project layer, a business logic layer, a data access layer, and a business object layer. The main project layer contains the '.aspx' files. The business logic layer provides all of the logical functionalities for the application. The layer works as a linking layer between the data access layer and the main project layer. The data access layer communicates with the business logic layer and collects data from the database. The business object layer contains objects and their values. Oracle 12C is used as the database.

4.2 System Features

The application has almost all functionality required by a question and answer (Q and A) forum. In addition to this, the application also shows overall and separate reputation scores for each category tag. This view helps the users to identify the best answer concerning the keywords and user. The main functionalities of the application are user creation, login, creating posts, viewing posts, viewing a single post with its answers, viewing reputation scores for every user and a user dashboard. The list of posts can be ordered in several ways, e.g., according to the latest posts, most viewed posts, most commented posts, or most favourites posts. The forum also has the feature to search posts by their titles and tags. The posts are listed in the form of a table, and each row consists of titles, contents, main category, and last three participant posts. Additionally, total

number of comments to the post, the total number of viewers, date/time when the post was create are also visible to the users. Users can click on each participants name and view their basic information (including the name of the participant, when they joined the platform, overall reputation scores, reputation scores per category and achieved badges). These attributes were chosen so as to provide users with an overall idea of who their co-members are, thereby assisting in establishing a sort of trustworthiness among members of the community.

Users can click on each post, which then opens the post as a separate page. The post page shows users the posted question, their answers, comments, and edit-options for each post. Each post itself contains the contributor's name, the date when the post was created, its description, up and down vote options, its count and on option to mark the post as favourite. In addition to the question post, there are also options to create answers, make edits and add comments to the post. On the same page, users can see the basic information about the contributor by clicking on the contributor's name. To create a new post, users can select the 'Create New Post' option from the provided menus. Under the 'Create New Post' form, user can add the title, main category, subcategory, description and also upload relevant documents. The options to select tags is provided in the main and subcategory fields. Under the subcategory field, user can select multiple categories, as per their convenience.

To reiterate, a key advantage of designed artifact is that users can view the overall and individual reputation of all their co-members. This would helps users identify the best answers/contributions. The application also has the option to give votes to the other users based on the posts/contributions and behaviour. The code for the designed artifact and the associated database files are openly available as a repo on GitHub (<https://github.com/ssijopious/CI-Forum>). This is done so that the results presented in the work, can be reproduced and built upon by others.

5 EVALUATION

In this section we attempt to answer the questions we raised previously in this work. The first question, how to implement CI methods in the BI platform so as to solve business organization's decision-making problems related to trust and sensemaking in the process of decision making.

As we eluded to earlier, BI systems can help resolve issues and support in the process of business organizational sensemaking and trust, however it there is a need to create crowd-based platforms to make ensure data quality, flexibility and risk management. And maintaining data quality, requires that the source of the data are given higher priority. To make sure the integrity of the source, we can utilize the collective knowledge of humans using crowd-sourcing methods within BI systems. To entrust a source or user, would require time, and trustable users would need to contribute trustworthy information while also cooperating with other users of the system. The continuous interactions of the user would help develops trust in the platform. This accuracy of trust will have a high impact on the business organizational decision-making processes.

To solve the next question, this study proposes a new reputation model to identify the problems of the CI reputation model and support the BI system to make more trust and sensible decisions. To evaluate this artifact quantitative research method is used. A

question and answer platform are created to implement this new reputation system (CI-Forum). A target group is selected for testing this platform and making the evaluations. In this evaluation, we tried to identify the target group's general understanding and habits of the reputation model. The target users are software engineers and IT specialists. Most of the participants have experience in using question and answer platform. The target group is from two different countries. To collect the evaluation, a questionnaire is created.

5.1 Experimental Procedure

To evaluate the designed artifact we conducted lab experiments with multiple users. The candidates for the experiments were identified through social media (primarily Facebook), by using snow-balling. More that 50 potential candidates were identified and given presentation on how to use the platform. After the presentation, the candidates (i.e., participants or users) were provided the web address of the application (which was hosted online during the experiments). Each participant was asked to create separate user profiles, and were instructed to create multiple posts (questions, answers and comments). After this, the participants were asked to actively use the platform over the next two weeks. It is important to note here that all participants had a background in software development, hence they were asked to use the platform in the daily workflows. At the end of two weeks, more than 75 questions with multiple answers had been posted on the platform.

After this, all participants were forwarded survey questionnaires, and were given two days to fill in the same. In total, 68 questionnaires were collected at the end of the experiment. Only 45 valid opinions we found, and hence the remaining were 23 questionnaire responses were rejected.

5.2 Reputation Model

To assess the reputation model, the participants we asked questions related to identification of trustable users. This included three questions (given below), and participants were asked to score the questions through Likert scale ranging from (1) indicating 'Completely Disagree' to (5) indicating 'Completely Agree'. The results of the participants feedback is illustrated in Table and Figure .

- Did the CI-forum help the participant to identify the trustworthy user?
- Did the CI-forum help to analyze user expertise?
- Did the CI-Forum provide more overview of the users?

The participants feedback illustrates that the proposed reputation model helped users in identify trustful users. By showing a separate reputation for each category, users were able to identify the area of expertise of their co-members. The platform also helped users gain a better overview of their co-members overall. As indicated in Table and Figure for every question, most of the participants voted for 'Agree' and the average score was more than 3, so we conclude that the reputation model successfully assists users in making sensible decisions through the use of reputation score. The overall score of 3.6 indicates that all participants agreed with the new reputation model approach and were ready to accept the reputation scores. If a user had a high reputation score, then

Figure 1: A screenshot of list of posts as viewed by end-users on the proposed CI-Forum

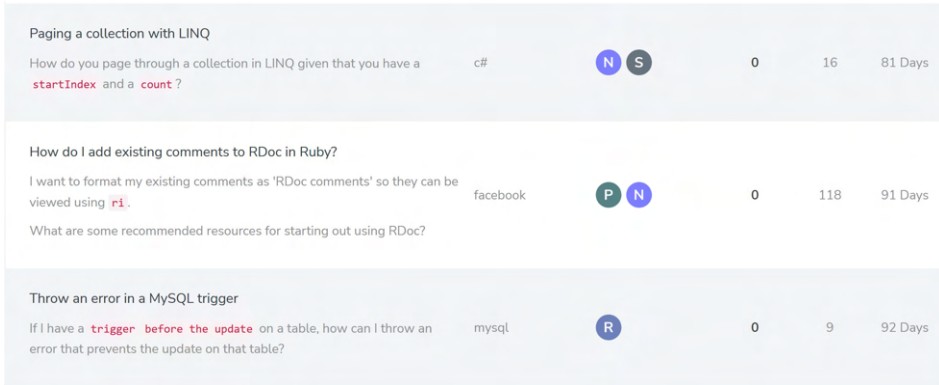


Table 1: User’s assessment of the Reputation Model

Sub-factors		Level of Agreements					Mean	
		1	2	3	4	5		
Trust	N	1	5	11	18	10	3.7	Agree
	%	2.2	11.1	24.4	40	22.2		
User Expertise	N	1	4	11	16	14	3.9	Agree
	%	2.2	8.9	24.4	35.6	31.4		
Overview of Users	N	2	4	19	19	2	3.3	Agree
	%	4.4	8.9	42.2	42.2	2.2		
<i>Total</i>		<i>4</i>	<i>13</i>	<i>41</i>	<i>53</i>	<i>26</i>	<i>3.6</i>	<i>Agree</i>

their fellow users considered them as a trustworthy users and accepted their answers. These results also answer the second research question raised in this work. We can create a reputation model to solve the trust problem in BI by showing separate reputation score for each category, as this method benefits users by helping them identify the experts and helps users select the best inputs according to this information. This further aids BI to maintain data quality thereby assisting in sensible decision-making. We argue that this approach compels users to contribute consistently and mimics reputation as it exists in the real-world.

5.3 Usability of CI-Forum

To assess usability, the questionnaire (presented to the participants) contained four questions all revolving around the systems user interface and features. Answers to these provide us an overview of user interactions and the usability and ease-of-use of the designed CI-Forum. These questions again were supposed to be answered using a Likert scale ranging from (1) indicating ‘Completely Disagree’ to (5) indicating ‘Completely Agree’.

- CI-Forum is easy to use or not?
- Are you willing to continue using the CI-Forum?
- Is CI-Forum having a clearer and easier operating interface?
- CI-Forum will be recommended to family and friends?

Table 2: User’s feedback regarding the usability of the proposed CI-Forum

Sub-factors		Level of Agreements					Mean	
		1	2	3	4	5		
Easy to use	N	2	7	17	16	3	3.2	Agree
	%	4.4	15.6	37.8	35.6	6.7		
Will continue to use	N	0	7	12	20	6	3.6	Agree
	%	0.0	15.6	26.7	44.4	13.3		
Easier operating interface	N	0	7	15	18	6	3.6	Agree
	%	0.0	15.6	33.3	40.0	13.3		
Recommended to family and friends	N	1	6	17	12	9	3.5	Agree
	%	2.2	13.3	37.8	26.7	20.0		
<i>Total</i>		<i>3</i>	<i>27</i>	<i>61</i>	<i>66</i>	<i>24</i>	<i>3.6</i>	<i>Agree</i>

As Table and Figure indicate, the users found the system’s interface easy-to-use and the forum in general usable. The users’ interaction with CI-Forum were meaningful as they did not face any issues while using the application. Most of the users stated that they would to continue as well as recommended to their friends and family. The mean value of every question was more than 3. The average of the mean value was 3.5, which means that all users were satisfied with their interactions with the CI-Forum. Most users agreed that CI-Forum is useful for their purposes.

During the development phase of the CI-forum, additional feedback was gathered from industry experts, especially those working in the field of software development and testing. These feedback were used to enhance the systems functionalities and usability. Most feedback gathered during this process was positive, and although the experiments with participants was carried out at a smaller sample size, almost all participants simulated actual real-world end-users the CI-forum is oriented towards—as mostly confirmed by

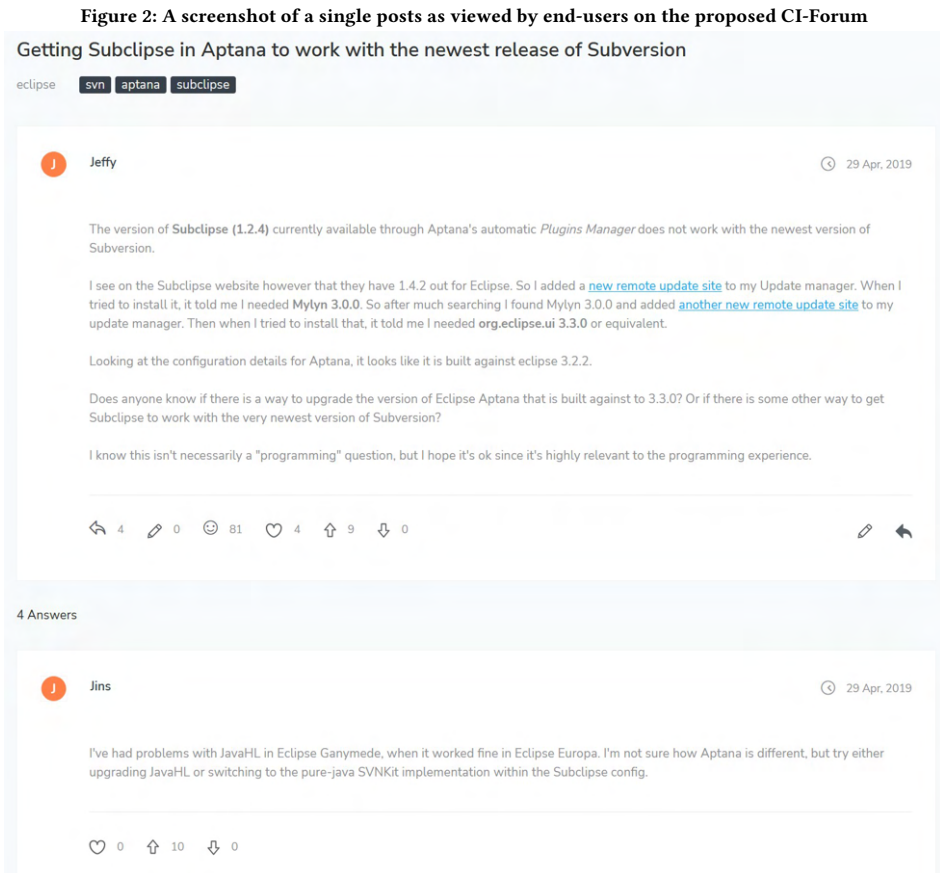
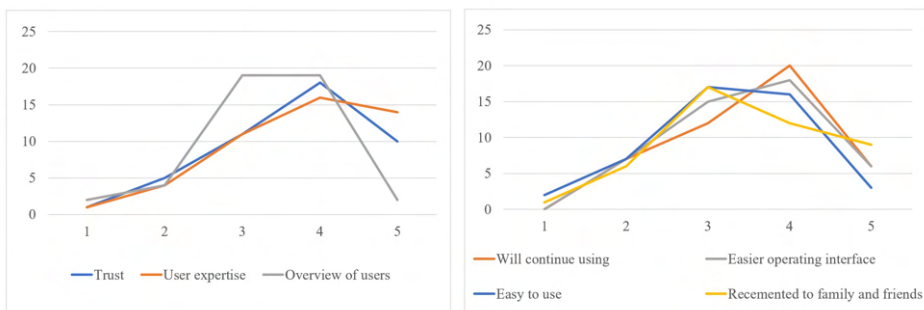


Figure 3: Users' evaluation of the proposed Reputation Model (left) and usability of CI-Forum (right)



the obtained results. The results of the experiments and its following quantitative analysis will be utilized in future to improve the CI-Forum further. The results of the above experiments are only limited by the number and homogeneity of the participant sample, and further user tests are required to develop more conclusive outcomes.

6 CONCLUSION

The overall aim of this work was two main challenges that are encountered when using BI strategies today, these are, sensemaking and trust. Given the critical nature of BI strategies in solving business organizational issues and in supporting organizational decision-making processes; we set out to solve the issues of sensemaking and trust by drawing influences from research in CI. We proposed a novel crowdsourcing approach to reputation models, built around a novel discussion-forum, with focus on organizational employees' perspective and helps establish trust among employees when using BI systems and strategies. By showing users separate reputation scores for each area of expertise, users were able to identify the experts among their fellow users. The idea, behind the approach was that if trustable users works together, the information and results generated by them would be by those organizations is more trustable and sensible to the organizations, specially when compared with non-expert/trustable employees.

The main challenges encountered in this work was that current technologies are still not well adapted to such scenarios. The evaluation of both the reputation model and the CI-forum were only carried out at a small scale, with limited number of participants. Hence the accumulated results only present a superficial view of the usability and usefulness of the proposed contributions. Further changes and fine-tuning is required to enhance the developed artifact. For now, the artifact allows users to identify users with expertise in specific tasks, however, for the next iteration of the forum we would like to develop it so that it can accommodate multi-organization scenarios. Also, as part of future work, we would like to investigate (on a larger scale) and understand the long-term effects of use of reputation scores within organizations and their BI systems.

ACKNOWLEDGEMENT

This work has been developed as part of the "ICT programme" project and was supported by the European Union through European Social Fund.

REFERENCES

- [1] Mohammad Allahbakhsh, Boualem Benatallah, Aleksandar Ignjatovic, Hamid Reza Motahari-Nezhad, Elisa Bertino, and Schahram Dustdar. 2013. Quality Control in Crowdsourcing Systems: Issues and Directions. *IEEE Internet Computing* 17, 2 (2013), 76–81. <https://doi.org/10.1109/MIC.2013.20>
- [2] B. Azvine, Z. Cui, D.D. Nauck, and B. Majeed. 2006. Real Time Business Intelligence for the Adaptive Enterprise. In *The 8th IEEE International Conference on E-Commerce Technology and The 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services (CEC/EEE'06)*. 29–29. <https://doi.org/10.1109/CEC-EEE.2006.73>
- [3] Richard J. Boland. 2008. *Decision Making and Sensemaking*. Springer Berlin Heidelberg, Berlin, Heidelberg, 55–63. https://doi.org/10.1007/978-3-540-48713-5_3
- [4] Efthimios Bothos, Dimitris Apostolou, and Gregoris Mentzas. 2012. Collective intelligence with web-based information aggregation markets: The role of market facilitation in idea management. *Expert Systems with Applications* 39, 1 (2012), 1333–1345. <https://doi.org/10.1016/j.eswa.2011.08.014>
- [5] Daren C. Brabham. 2008. Crowdsourcing as a Model for Problem Solving: An Introduction and Cases. *Convergence* 14, 1 (2008), 75–90. <https://doi.org/10.1177/1354856507084420>
- [6] Daren C. Brabham. 2013. *Crowdsourcing*. MIT Press. <https://mitpress.mit.edu/books/crowdsourcing>
- [7] Thierry Buecheler, Jan Henrik Sieg, Rudolf Marcel Fuchsli, and Rolf Pfeifer. 2010. Crowdsourcing, open innovation and collective intelligence in the scientific method: a research agenda and operational framework. In *The 12th International Conference on the Synthesis and Simulation of Living Systems, Odense, Denmark, 19–23 August 2010*. MIT Press, 679–686. <https://doi.org/10.21256/zhaw-4094>
- [8] Hsinchun Chen, Roger H. L. Chiang, and Veda C. Storey. 2012. Business Intelligence and Analytics: From Big Data to Big Impact. *MIS Quarterly* 36, 4 (2012), 1165–1188. <http://www.jstor.org/stable/41703503>
- [9] Xiaowen Chu, Xiaowei Chen, Kaiyong Zhao, and Jiangchuan Liu. 2010. Reputation and trust management in heterogeneous peer-to-peer networks. *Telecommunication Systems* 44, 3 (01 Aug 2010), 191–203. <https://doi.org/10.1007/s11235-009-9259-5>
- [10] Thomas D. Clark, Mary C. Jones, and Curtis P. Armstrong. 2007. The Dynamic Structure of Management Support Systems: Theory Development, Research Focus, and Direction. *MIS Quarterly* 31, 3 (2007), 579–615. <http://www.jstor.org/stable/25148808>
- [11] Ian J. Deary. 2012. Intelligence. *Annual Review of Psychology* 63, 1 (2012), 453–482. <https://doi.org/10.1146/annurev-psych-120710-100353> arXiv:<https://doi.org/10.1146/annurev-psych-120710-100353> PMID: 21943169.
- [12] Pierpaolo Dondio and Luca Longo. 2011. Trust-Based Techniques for Collective Intelligence in Social Search Systems. In *Next Generation Data Technologies for Collective Computational Intelligence*, Nik Bessis and Fatos Xhafa (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 113–135. https://doi.org/10.1007/978-3-642-20344-2_5
- [13] R. Duane Ireland and Justin W. Webb. 2007. Strategic entrepreneurship: Creating competitive advantage through streams of innovation. *Business Horizons* 50, 1 (2007), 49–59. <https://doi.org/10.1016/j.bushor.2006.06.002>
- [14] Colette Dumas. 2010. Hosting Conversations for Effective Action. *Journal of Knowledge Globalization* 3, 1 (2010), 99 – 116. <https://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=51902286&site=eds-live>
- [15] Mark Dworkin, Lois Foreman-Wernet, and Brenda Dervin. 1999. Sense-Making and television news: An inquiry into audience interpretations. *The Electronic Journal of Communication* 9, 2 (1999). <http://www.cios.org/EJCPUBLIC/009/2/009217.html>
- [16] Diego Gambetta et al. 2000. Can we trust trust. *Trust: Making and breaking cooperative relations* 13 (2000), 213–237.
- [17] John Hancock and Roger Toren. 2006. *Practical Business Intelligence with Sql Server 2005* (first ed.). Addison-Wesley Professional.
- [18] Borut Hocevar and Jurij Jaklič. 2010. Assessing benefits of business intelligence systems—a case study. *Management: journal of contemporary management issues* 15, 1 (2010), 87–119. <https://hrcaek.srce.hr/53609>
- [19] Kevin Hoffman, David Zage, and Cristina Nita-Rotaru. 2009. A Survey of Attack and Defense Techniques for Reputation Systems. *ACM Comput. Surv.* 42, 1, Article 1 (Dec. 2009), 31 pages. <https://doi.org/10.1145/1592451.1592452>
- [20] Steve LaValle, Eric Lesser, Rebecca Shockley, Michael S Hopkins, and Nina Kruschwitz. 2011. Big data, analytics and the path from insights to value. *MIT sloan management review* 52, 2 (2011), 21–32. <https://sloanreview.mit.edu/article/big-data-analytics-and-the-path-from-insights-to-value/>
- [21] Thomas W. Malone and Michael S. Bernstein (Eds.). 2015. *Handbook of Collective Intelligence*. The MIT Press, Cambridge, MA, 230 pages.
- [22] Ojelanki K. Ngwenyama and Noel Bryson. 1999. Making the information systems outsourcing decision: A transaction cost approach to analyzing outsourcing decision problems. *European Journal of Operational Research* 115, 2 (1999), 351–367. [https://doi.org/10.1016/S0377-2217\(97\)00171-9](https://doi.org/10.1016/S0377-2217(97)00171-9)
- [23] Muhammad I. Nofal and Zawiya M. Yusof. 2013. Integration of Business Intelligence and Enterprise Resource Planning within Organizations. *Procedia Technology* 11 (2013), 658–665. <https://doi.org/10.1016/j.protcy.2013.12.242> 4th International Conference on Electrical Engineering and Informatics, ICEEI 2013.
- [24] O Pilli. 2014. LMS Vs. SNS: Can social networking sites act as a learning management systems. *American International Journal of Contemporary Research* 4, 5 (2014), 90–97. http://www.ajcernet.com/journals/Vol_4_No_5_May_2014/9.pdf
- [25] V. Pirttimaki. 2007. Conceptual analysis of business intelligence. *SA Journal of Information Management* 9, 2 (2007). <https://doi.org/10.4102/sajim.v9i2.24>
- [26] Mahesh S Raisinghani. 2003. *Business Intelligence in the Digital Economy: Opportunities, Limitations and Risks: Opportunities, Limitations and Risks*. Idea Group Pub. <https://books.google.ee/books?id=xKszZbc7RhYc>
- [27] Paul Resnick and Richard Zeckhauser. 2002. Trust among strangers in internet transactions: Empirical analysis of eBay's reputation system. *Advances in Applied Microeconomics*, Vol. 11. Emerald Group Publishing Limited, 127–157. [https://doi.org/10.1016/S0278-0984\(02\)11030-3](https://doi.org/10.1016/S0278-0984(02)11030-3)
- [28] J.P. Shim, Merrill Warkentin, James F. Courtney, Daniel J. Power, Ramesh Sharda, and Christer Carlsson. 2002. Past, present, and future of decision support technology. *Decision Support Systems* 33, 2 (2002), 111–126. [https://doi.org/10.1016/S0278-0984\(02\)11030-3](https://doi.org/10.1016/S0278-0984(02)11030-3)

- [//doi.org/10.1016/S0167-9236\(01\)00139-7](https://doi.org/10.1016/S0167-9236(01)00139-7) Decision Support System: Directions for the Next Decade.
- [29] Shweta Suran, Vishwajeet Pattanaik, and Dirk Draheim. 2020. Frameworks for Collective Intelligence. *Comput. Surveys* 53, 1 (may 2020), 1–36. <https://doi.org/10.1145/3368986>
- [30] Shweta Suran, Vishwajeet Pattanaik, Sadok Ben Yahia, and Dirk Draheim. 2019. Exploratory Analysis of Collective Intelligence Projects Developed Within the EU-Horizon 2020 Framework. In *Computational Collective Intelligence*, Ngoc Thanh Nguyen, Richard Chbeir, Ernesto Exposito, Philippe Aniorté, and Bogdan Trawiński (Eds.). Springer International Publishing, Cham, 285–296.
- [31] Carlo Vercellis. 2011. *Business intelligence: data mining and optimization for decision making*. Wiley Online Library. <https://bit.ly/3FrU9fZ>
- [32] C. Nadine Wathen and Jacquelyn Burkell. 2002. Believe it or not: Factors influencing credibility on the Web. *Journal of the American Society for Information Science and Technology* 53, 2 (2002), 134–144. <https://doi.org/10.1002/asi.10016>
- [33] Hugh J. Watson and Barbara H. Wixom. 2007. The Current State of Business Intelligence. *Computer* 40, 9 (2007), 96–99. <https://doi.org/10.1109/MC.2007.331>
- [34] K.E. Weick. 2012. *Making Sense of the Organization, Volume 2: The Impermanent Organization*. Wiley. <https://bit.ly/3oEHY4>
- [35] K.E. Weick and K.E.W. Weick. 1995. *Sensemaking in Organizations*. SAGE Publications. <https://bit.ly/3BkaKzT>
- [36] Michael Weiss. 2016. Crowdsourcing Literature Reviews in New Domains. *Technology Innovation Management Review* 6 (02/2016 2016), 5–14. <https://doi.org/10.22215/timreview/963>
- [37] Sean Wise, Robert A. Paton, and Thomas Gegenhuber. 2012. Value co-creation through collective intelligence in the public sector. *VINE* 42, 2 (01 Jan 2012), 251–276. <https://doi.org/10.1108/03055721211227273>
- [38] Barbara Wixom and Hugh Watson. 2010. The BI-Based Organization. *International Journal of Business Intelligence Research (IJBIR)* 1, 1 (2010), 13–28. <https://doi.org/10.4018/ijbir.2010071702>
- [39] Öykü Işık, Mary C. Jones, and Anna Sidorova. 2013. Business intelligence success: The roles of BI capabilities and decision environments. *Information & Management* 50, 1 (2013), 13–23. <https://doi.org/10.1016/j.im.2012.12.001>

Appendix 7

Getting Started Guide for Tippanee

Tippaneer – Getting Started Guide

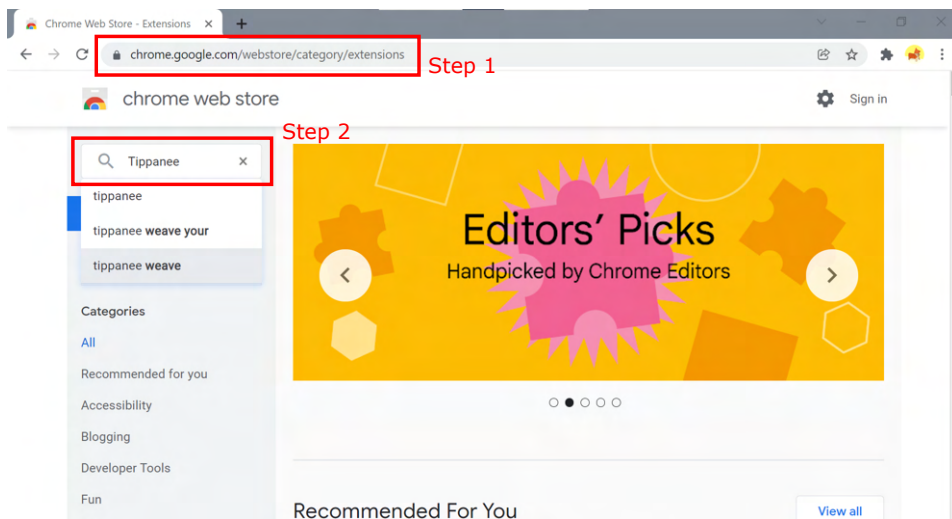
Introduction

The following guide is meant to enable users to install and start creating annotations with Tippaneer. The primary features and user-interface components of the annotation tool are demonstrated through screenshots captured on a Windows machine.

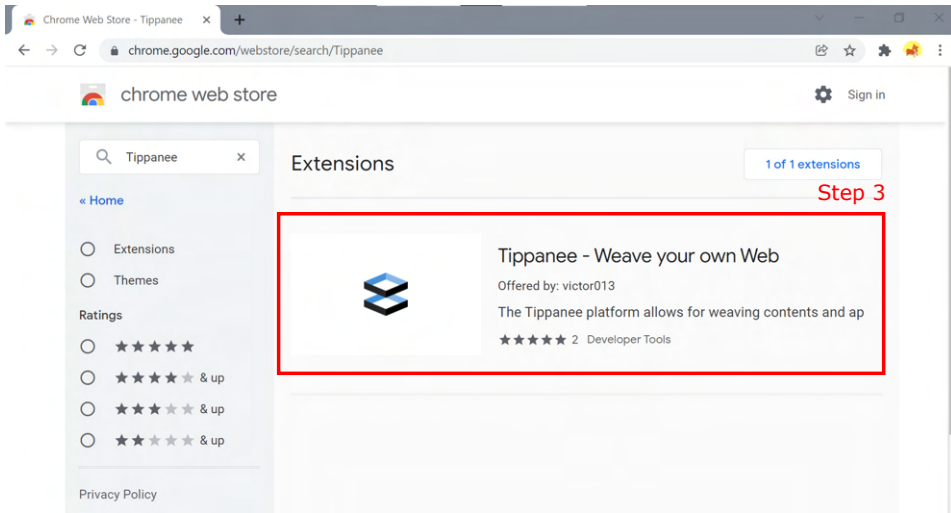
Installing Tippaneer

To begin, we will need to first install Tippaneer’s browser extension on the Google Chrome browser.

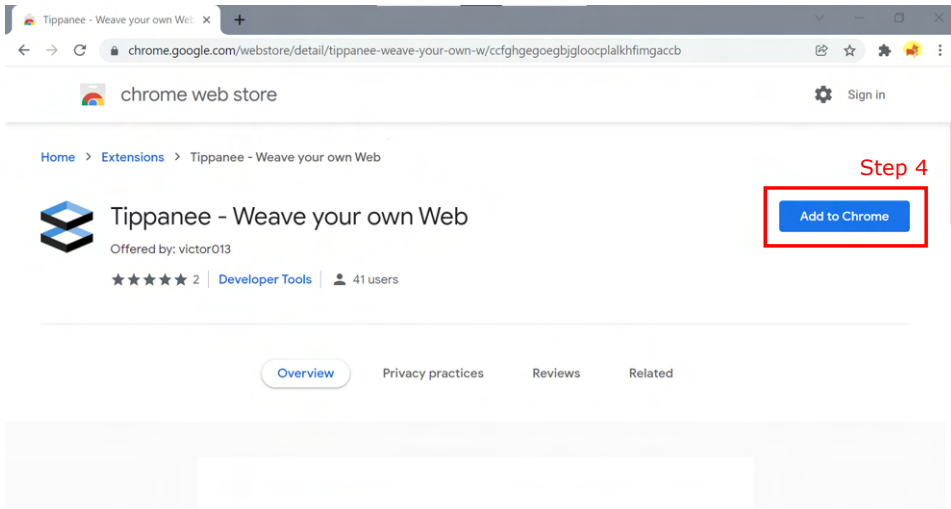
1. Start by opening the Google Chrome browser on your system, and then go to the *Chrome Web Store*. To get to the *Chrome Web Store*, type the URL: “https://chrome.google.com/webstore/category/extensions” in the address bar of the browser, and hit the *Enter* key.
2. Once on the *Chrome Web Store*, type the word “Tippaneer” on the search bar on the left, and hit the *Enter* key.



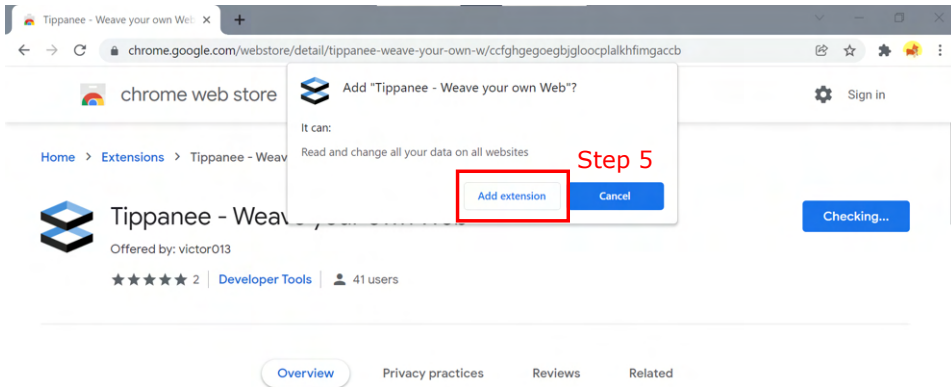
3. The Tippianee browser extension should now be visible at the center of screen. Click on the extension description, to get to the extension's page.



4. The extension's page should have an *Add to Chrome* button on the right side of the page. Click on the button to install the extension on to your browser.

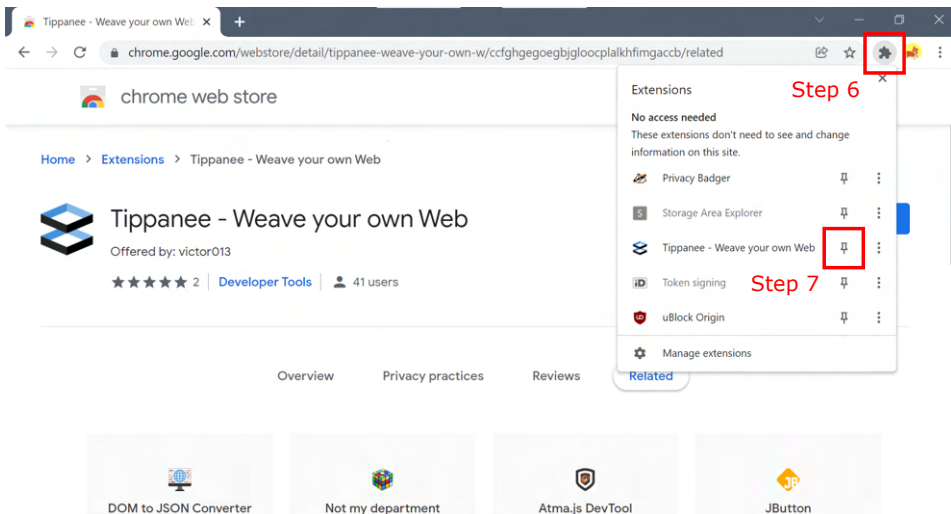


5. You will be prompted with the permissions required by the extension. Click the *Add extension* button to go ahead with the installation.

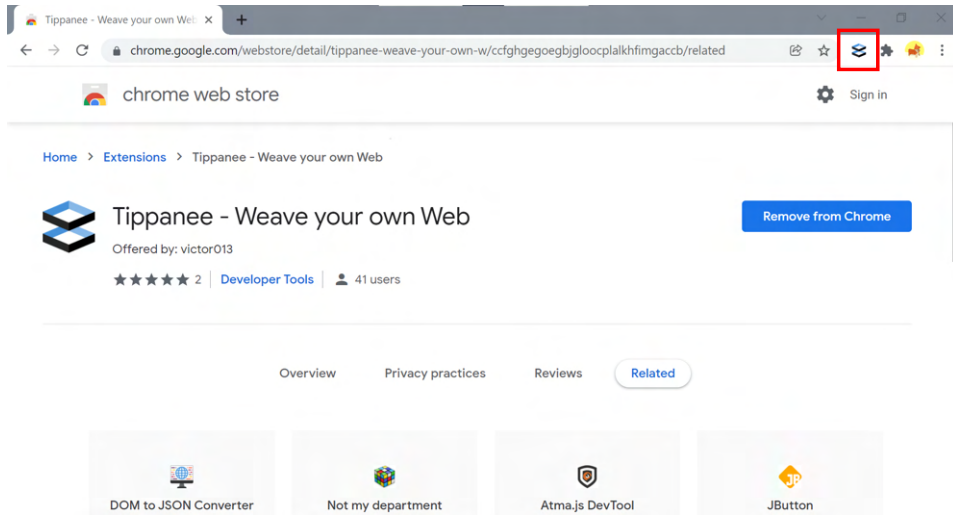


6. You will be prompted, once the extension is installed. Now, to view the extension shortcut on your browser, click on the puzzle-like logo at the top-right corner of the browser window. This *Extension* button allows users to pin extensions to the browser UI.

7. Click on the pin, to the right of the extension "Tippane - Weave your own Web".



The Tippane extension should now be ready for use, and the extension's shortcut should be visible in the top-right area of the browser window.



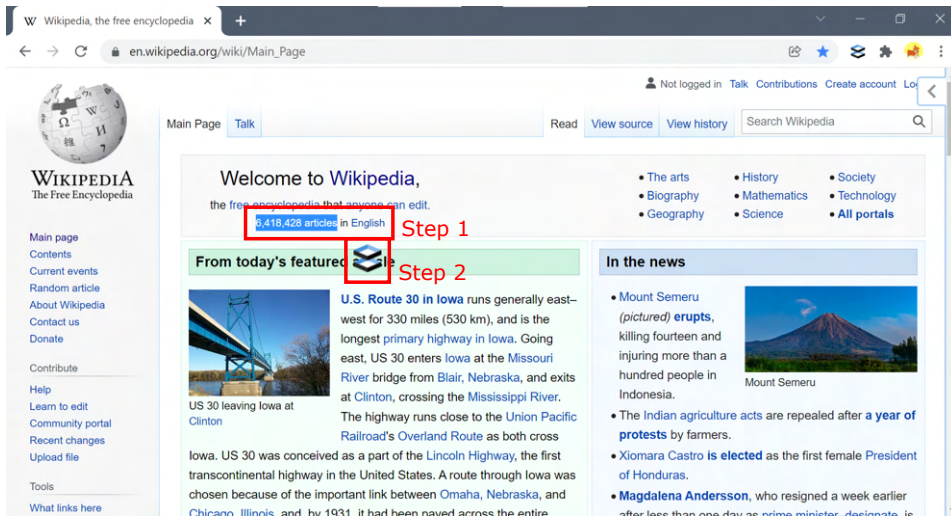
To enable or disable the extension on any web page, click on the extension shortcut once. If Tippane's extension logo turns to gray-scale, this means the extension is disabled. While if the extension logo is colored (i.e., black, blue and white), this implies the extension is active. Please note, that if the extension takes to long to load on a page, or doesn't seem to work, you should disable and re-enable the extension.

Creating Annotations

To create an annotation on Tippaneer, simply visit the web you would like to annotate, verify whether the extension is enabled on the web page, and then follow these steps:

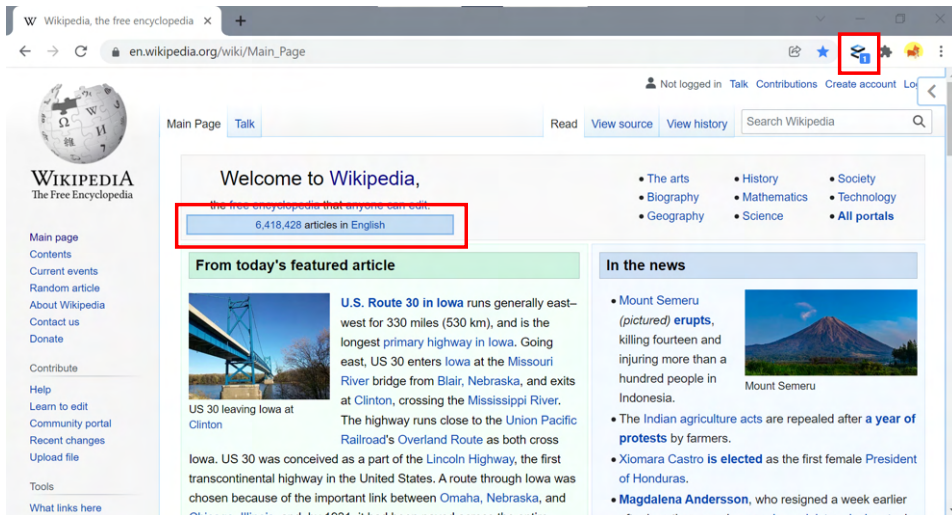
Please note that we demonstrate these steps by creating annotations on the Wikipedia home page (https://en.wikipedia.org/wiki/Main_Page).

1. Select the text you would like to annotate, by clicking and dragging the mouse over the text. Once the text is selected you should see Tippaneer's logo to the bottom-right of the selected text.
2. Click on the Tippaneer logo to annotate the selected text.



The screenshot shows the Wikipedia Main Page in a browser window. The address bar displays "en.wikipedia.org/wiki/Main_Page". The page content includes a "Welcome to Wikipedia" section with the text "the free encyclopedia that anyone can edit." and "5,418,428 articles in English". A red box highlights the text "5,418,428 articles in English" with the label "Step 1". Below this, a green box highlights the "From today's featured" section with the label "Step 2". The "From today's featured" section includes a photo of a bridge and text about "U.S. Route 30 in Iowa". The "In the news" section lists several news items, including "Mount Semeru (pictured) erupts" and "The Indian agriculture acts are repealed after a year of protests".

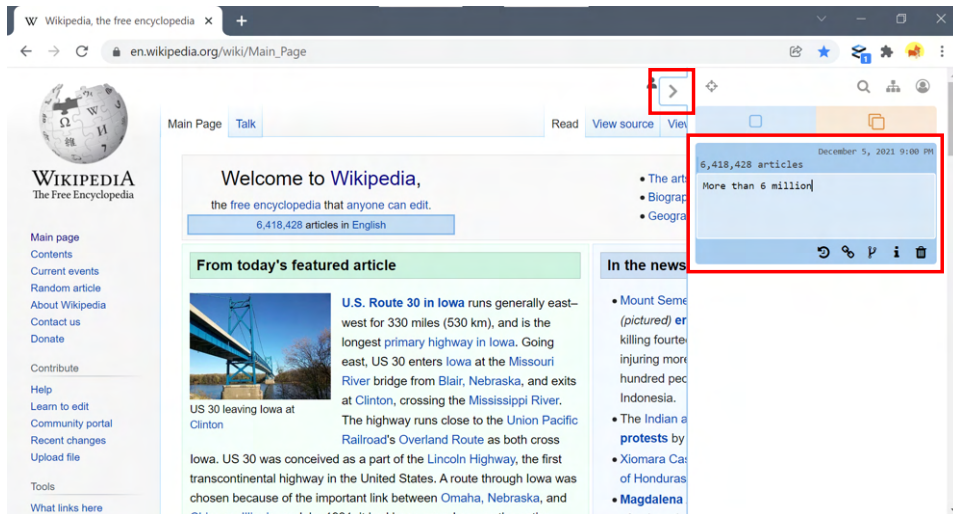
Once the text is annotated, you should see a light-blue box around the previously selected text. The box indicates the HTML DOM element that the annotated text belongs to. The Tippane shortcut in the top-right area of the browser indicates the number of annotations of the current web page.



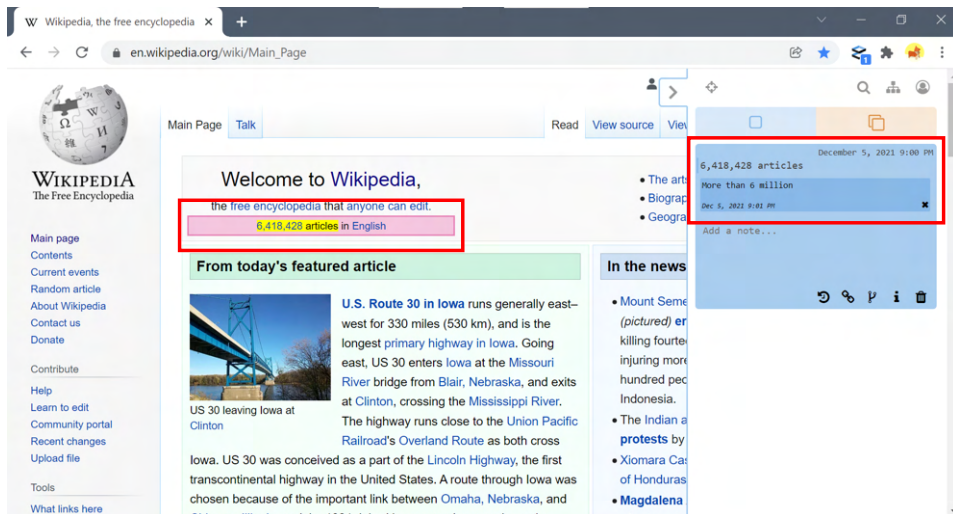
Viewing Annotations and Adding Comments

To view the annotations previously created on a web page, click the left/right-facing arrow on the right side of the browser. By default the dashboard of the extension is hidden. However clicking on the left-facing arrow, should make the dashboard visible on the right side of the browser window.

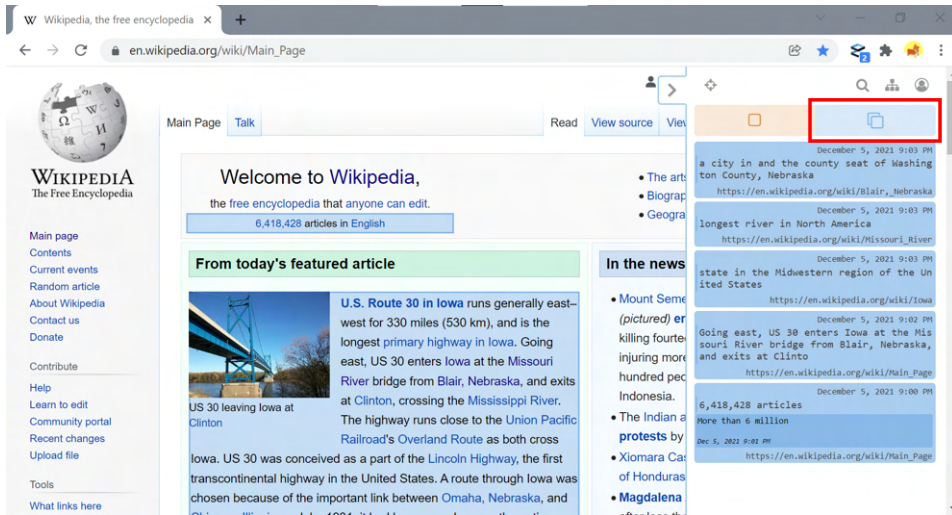
To add comments to an annotation, click on the text box with the message “Add a note...”. Then type in comment you would like to add, and hit the *Enter* key.



To view an annotation on the web page, click on the annotation within the dashboard. Doing so, should highlight the exact annotated text in yellow, and the annotated element in pink.



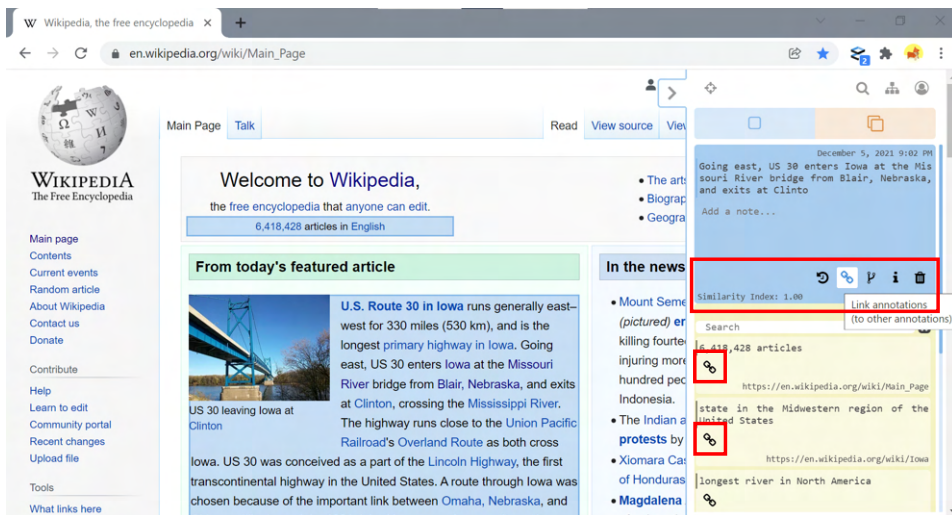
To view all annotations you have created so far, click on the *Browse all annotations* button, indicated by two overlapping squares, inside the annotation tool's dashboard.



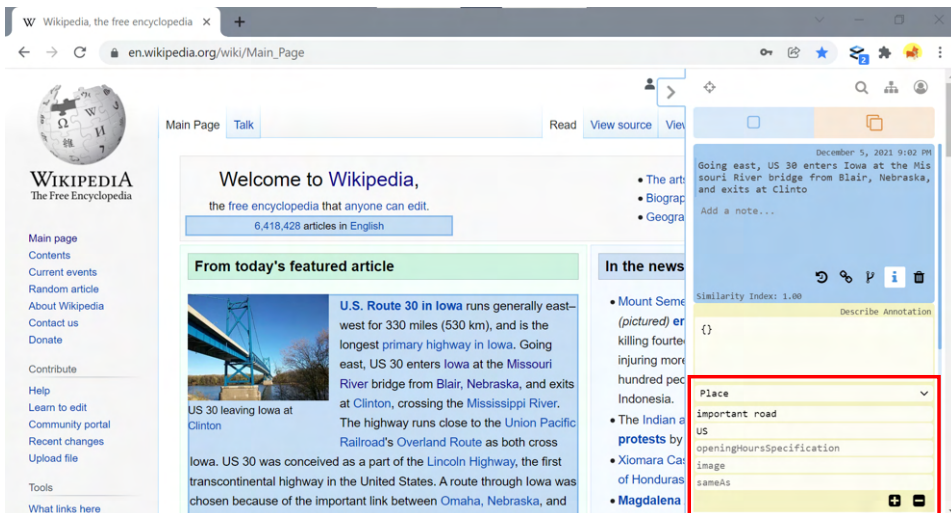
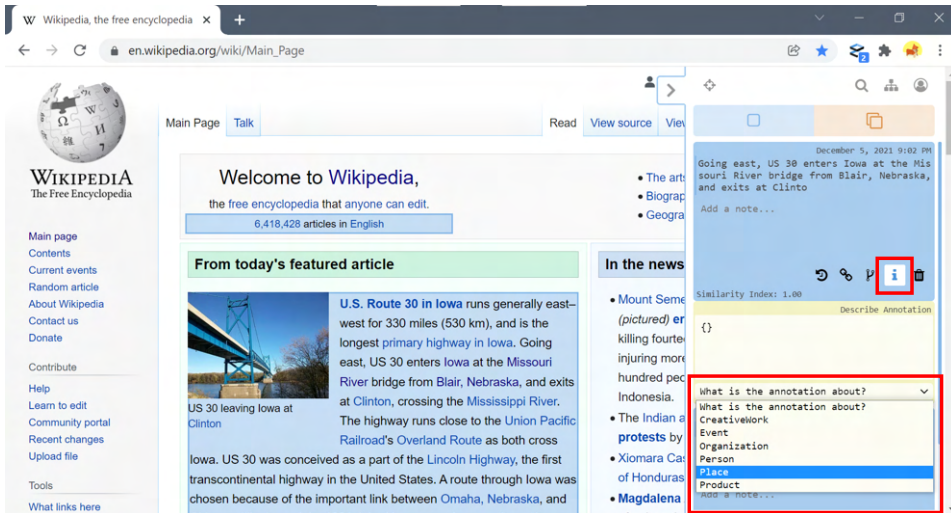
User Interface and System Features

If a user visits a previously annotated web page, the Tippane extension uses its novel anchoring algorithm and attempts to reattach the annotations to their correct locations. In doing so, the extension generates a *similarity index* for each of the page's annotations. The *similarity index* of the reattached annotation can be found on the bottom-left of the corresponding annotation, inside the annotation tool's dashboard. To the right of the *similarity index* indicator, are the buttons: *Reconstruct annotation*, *Link annotations*, *Transclude annotation*, *Describe annotation*, and *Delete annotation*.

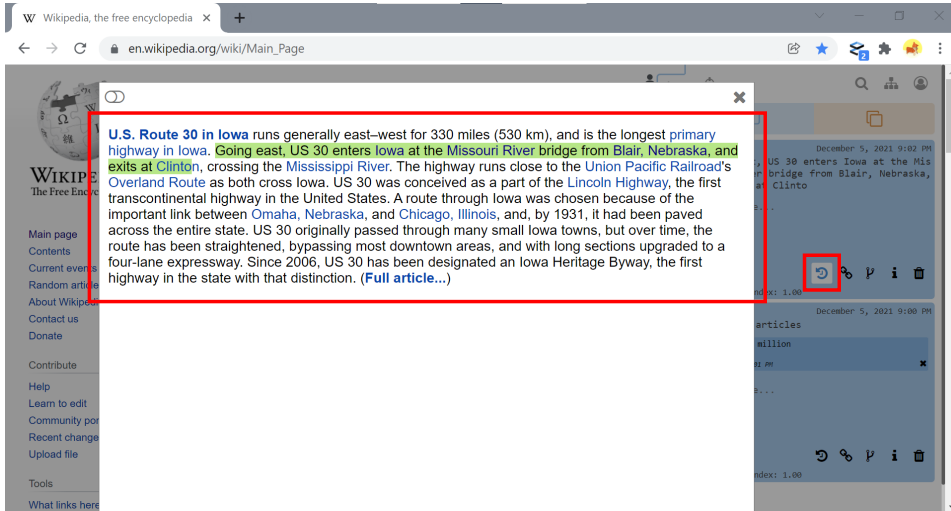
To create links between annotations, first click on the *Link annotations* button. A list of all annotations created by the user should appear under the annotation. By clicking on *Add link* button, indicated by the link logo, a user can create a uni-direction link between the original annotation and the annotation being linked. By clicking on the same logo again user can unlink the annotations (i.e., *Remove link*).



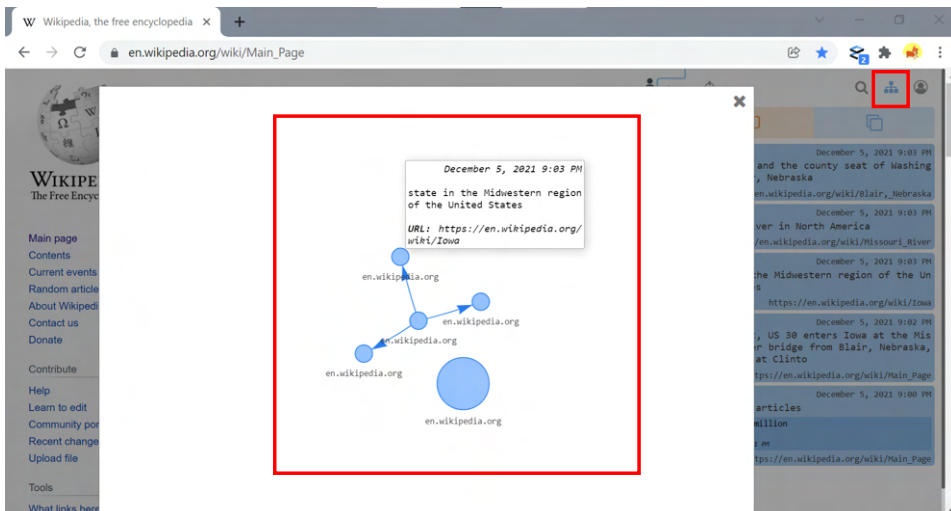
To add semantic descriptions to annotated contents, users can use the *Describe annotation* button. On clicking the button, users should be able to see the *Describe Annotation* window, under the original annotation. The new window contains a drop-down menu titled “What is the annotation about?”. Users can choose between a few pre-defined semantic classes and properties, that they can then use to describe the context of the annotation. Once the users have added information into the (semantic metadata) text boxes, they are required to click on the *Add* button to store the added metadata. For now, this information is used to retrieve annotations, when users are searching for a specific annotation, using the *Search annotations* feature.



Using Tippane's *Reconstruct annotation* feature, users can view created annotations in their original forms. The feature works by showing users how the annotation looked like when it was first created. Also, it allows users to see both the annotated content and its surrounding context, given the annotation is part of a larger DOM element. The annotated text is made visible with a green highlight, while its surrounding text has no highlight.



Users can view linked annotations by using the *Visualize annotations* feature. Activated by using the tree-like logo in the top-right area of the annotation tool dashboard, the feature allows users to visualize the annotations and their links. Through the feature, annotations are visualized as a node, while their links are visualized as edges. It should be noted that the size of each node indicates the number of comments the annotation has. By hovering the mouse over a node, users can view the node's corresponding annotation text, while double-clicking on the node redirects users to the corresponding annotation's web page.



Curriculum Vitae

1. Personal data

Name Vishwajeet Pattanaik
Date and place of birth 13 July 1988, Odisha, India
Nationality Indian

2. Contact information

Address Tallinn University of Technology, School of Information Technologies, Department of Software Science, Akadeemia tee 15a, 12618 Tallinn, Estonia
Phone +372 5838 3989
E-mail vishwajeet.pattanaik@taltech.ee

3. Education

2017–... Tallinn University of Technology, School of Information Technologies, Computer Science, Doctor of Philosophy
2012–2014 SRM University, School of Computing, Knowledge Engineering, M.Tech
2007–2011 Dr. A.P.J. Abdul Kalam Technical University, Department of Computer Science, Computer Science Engineering, B.Tech

4. Language competence

Hindi native
English fluent
Odia basic level

5. Professional employment

2021– ... Centre for Biorobotics, Tallinn University of Technology, Engineer
2017–2021 Information Systems Group, Tallinn University of Technology, Early Stage Researcher
2016–2016 Indian Institute of Technology Delhi, Research Fellow
2014–2017 Krishna Engineering College (Dr. A.P.J. Abdul Kalam University), Assistant Professor
2013–2014 SRM Institute of Science and Technology (SRM University), Teaching Assistant
2011–2011 iYogi Technical Services Pvt. Ltd., Technical Specialist
2010–2010 NIIT Ltd., Technical Trainer

6. Voluntary work

2020–... Technology evangelist and mentor on several Discord servers

7. Computer skills

- Operating systems: MS Windows, GNU/Linux
- Document preparation: MS Office, Libre Office, Latex, VS Code
- Programming languages: Python, Bash, C, C++, Java
- Scientific packages: MATLAB, R, Jupyter, TensorFlow
- Others: Inkscape, Meshroom, Blender

8. Honours and awards

- 2014, Secured Gold medal for paper presentation during Research Day at SRM University, Tamil Nadu, India

9. Defended theses

- 2014, "Inducing Human-like Motion in Robots", M.Tech, supervisor Prof. Dr. S. Prabakaran, SRM Institute of Science and Technology (SRM University), School of Computing

10. Field of research

- Collective Intelligence
- Web Development
- Knowledge Engineering
- Machine Learning
- Digital Image Processing

11. Scientific work

Papers

1. V. Pattanaik, S. Suran, and S. Prabakaran. Inducing human-like motion in robots. In *Proceedings of the 6th IBM Collaborative Academia Research Exchange Conference (I-CARE) on I-CARE 2014*, I-CARE 2014, page 1–3, New York, NY, USA, 2014. Association for Computing Machinery
2. S. Suran, V. Pattanaik, and D. Malathi. Discovering shortest path between points in cerebrovascular system. In *Proceedings of the 6th IBM Collaborative Academia Research Exchange Conference (I-CARE) on I-CARE 2014*, I-CARE 2014, page 1–3, New York, NY, USA, 2014. Association for Computing Machinery
3. A. Gupta and V. Pattanaik. Survey paper on encryption, authentication and auditing services for better cloud security. *International Journal of Computer Applications*, 138(10):33–37, Mar. 2016
4. H. Tyagi, S. Suran, and V. Pattanaik. Weather - temperature pattern prediction and anomaly identification using artificial neural network. *International Journal of Computer Applications*, 140(3):15–21, Apr. 2016

5. V. Pattanaik, M. Singh, P. Gupta, and S. Singh. Smart real-time traffic congestion estimation and clustering technique for urban vehicular roads. In *2016 IEEE Region 10 Conference (TENCON)*, pages 3420–3423, 2016
6. A. Gupta, V. Pattanaik, and M. Singh. Enhancing k means by unsupervised learning using PSO algorithm. In *2017 International Conference on Computing, Communication and Automation (ICCCA)*. IEEE, May 2017
7. S. Suran, V. Pattanaik, M. Singh, P. Gupta, and P. Gupta. Brain imaging procedures and surgery techniques: Past, present and future. *International Journal of Bio-Science and Bio-Technology*, 9(3):23–34, June 2017
8. V. Pattanaik, A. Norta, M. Felderer, and D. Draheim. Systematic support for full knowledge management lifecycle by advanced semantic annotation across information system boundaries. In J. Mendling and H. Mouratidis, editors, *Information Systems in the Big Data Era*, pages 66–73, Cham, 2018. Springer International Publishing
9. V. Pattanaik, S. Suran, and D. Draheim. Enabling social information exchange via dynamically robust annotations. In *Proceedings of the 21st International Conference on Information Integration and Web-Based Applications & Services, iiWAS2019*, page 176–184, New York, NY, USA, 2019. Association for Computing Machinery
10. V. Pattanaik, I. Sharvadze, and D. Draheim. Framework for peer-to-peer data sharing over web browsers. In T. K. Dang, J. Küng, M. Takizawa, and S. H. Bui, editors, *Future Data and Security Engineering*, pages 207–225, Cham, 2019. Springer International Publishing
11. S. Suran, V. Pattanaik, S. B. Yahia, and D. Draheim. Exploratory analysis of collective intelligence projects developed within the eu-horizon 2020 framework. In N. T. Nguyen, R. Chbeir, E. Exposito, P. Aniorté, and B. Trawiński, editors, *Computational Collective Intelligence*, pages 285–296, Cham, 2019. Springer International Publishing
12. V. Pattanaik, I. Sharvadze, and D. Draheim. A peer-to-peer data sharing framework for web browsers. *SN Computer Science*, 1(4), June 2020
13. S. Suran, V. Pattanaik, and D. Draheim. Frameworks for collective intelligence: A systematic literature review. *ACM Comput. Surv.*, 53(1), Feb. 2020
14. S. Suran, V. Pattanaik, and D. Draheim. Communitycare: Tackling mental health issues with the help of community. In *Proceedings of the 22nd International Conference on Information Integration and Web-Based Applications & Services, iiWAS '20*, page 377–382, New York, NY, USA, 2020. Association for Computing Machinery
15. V. Dwivedi, V. Pattanaik, V. Deval, A. Dixit, A. Norta, and D. Draheim. Legally enforceable smart-contract languages: A systematic literature review. *ACM Comput. Surv.*, 54(5), June 2021
16. S. A. Peious, S. Suran, V. Pattanaik, and D. Draheim. Enabling sensemaking and trust in communities: An organizational perspective. In *Proceedings of the 23rd International Conference on Information Integration and Web-Based Applications & Services, iiWAS '21*, page 1–9, New York, NY, USA, 2021. Association for Computing Machinery

Conference presentations

1. V. Pattanaik. *Leveraging the Power of the Crowd to Save the Web*, Decentralised Web Symposium 2020: 17 January 2020, Vienna, Austria
2. V. Pattanaik, S. Suran, and D. Draheim. *Enabling Social Information Exchange via Dynamically Robust Annotations*, iiWAS 2019, 2–4 December 2019, Munich, Germany
3. V. Pattanaik, I. Sharvadze , and D. Draheim. *Framework for Peer-to-Peer Data Sharing over Web Browsers*, FDSE 2019, 27—29 November 2019, Nha Trang City, Vietnam
4. V. Pattanaik, A. Norta, M. Felderer , and D. Draheim. *Systematic Support for Full Knowledge Management Lifecycle by Advanced Semantic Annotation Across Information System Boundaries*, CAiSE 2018, 11–15 June 2019, Tallinn, Estonia
5. V. Pattanaik, M. Singh, P. K. Gupta and S. K. Singh. *Smart real-time traffic congestion estimation and clustering technique for urban vehicular roads*, TENCON 2016, 22–25 November 2016, Singapore
6. V. Pattanaik, S. Suran, and S. Prabakaran. *Inducing Human-like Motion in Robots*, I-CARE 2014, 9–11 October 2014, Bangalore, India

Elulookirjeldus

1. Isikuandmed

Nimi	Vishwajeet Pattanaik
Sünniaeg ja -koht	13 Juuli 1988, Odisha, India
Kodakondsus	India

2. Kontaktandmed

Adress	Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, Tarkvarateaduse instituut, Akadeemia tee 15a, 12618 Tallinn, Estonia
Telefon	+372 5838 3989
E-post	vishwajeet.pattanaik@taltech.ee

3. Haridus

2017–...	Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, Informaatika, Filosoofiadoktor
2012–2014	SRM University, School of Computing, Knowledge Engineering, M.Tech
2007–2011	Dr. A.P.J. Abdul Kalam Technical University, Department of Computer Science, Computer Science Engineering, B.Tech

4. Keelteoskus

hindi keel	emakeel
inglise keel	kõrgtase
odia keel	põhitase

5. Teenistuskäik

2021– ...	Biorobootika Keskus, Tallinna Tehnikaülikool, Insener
2017–2021	Infosüsteemide Töörühm, Tallinna Tehnikaülikool, Doktorant-nooremteadur
2016–2016	Indian Institute of Technology Delhi, Research Fellow
2014–2017	Krishna Engineering College (Dr. A.P.J. Abdul Kalam University), Assistant Professor
2013–2014	SRM Institute of Science and Technology (SRM University), Teaching Assistant
2011–2011	iYogi Technical Services Pvt. Ltd., Technical Specialist
2010–2010	NIIT Ltd., Technical Trainer

6. Vabatahtlik töö

2020–...	Tehnoloogia populariseerija ja mentor mitmes Discordi serveris
----------	----------------------------------------------------------------

7. Arvutioskused

- Operatsioonisüsteemid: MS Windows, GNU/Linux
- Kontoritarkvara: MS Office, Libre Office, Latex, VS Code
- Programmeerimiskeeled: Python, Bash, C, C++, Java
- Teadustarkvara paketid: MATLAB, R, Jupyter, TensorFlow
- Teised: Inkscape, Meshroom, Blender

8. Autasud

- 2014, kuldmedal ettekande eest teaduspäeval SRMi ülikoolis, Tamil Nodus, Indias.

9. Kaitstud lõputööd

- 2014, Inducing Human-like Motion in Robots, M.Tech, juhendaja Prof. Dr. S. Prbakaran, SRM Institute of Science and Technology (SRM University), School of Computing

10. Teadustöö põhisuunad

- Kollektiivne intelligentsus
- Veebiarendus
- Teadmistepõhine inseneriteadus
- Masinõpe
- Digitaalne pilditöötlus

11. Teadustegevus

Teadusartiklite, konverentsiteeside ja konverentsiettekannete loetelu on toodud ingliskeelse elulookirjelduse juures.

ISSN 2585-6901 (PDF)
ISBN 978-9949-83-869-1 (PDF)