TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Reimo Vellemaa 162943IAPM

A Lexicon of Semantic Similarity and Difference: An Analysis of the 20 000 Most Common English Words

Master's thesis

Supervisors: Martin Verrev MSc,
Tanel Tammet PhD

Tallinn 2023

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Reimo Vellemaa 162943IAPM

Semantilise sarnasuse ja eristatavuse leksikon: Analüüs 20 000 kõige levinumast ingliskeelsest sõnast

Magistritöö

Juhendajad: Martin Verrev MSc,
Tanel Tammet PhD

Tallinn 2023

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Reimo Vellemaa

Abstract

This thesis aims to address the lack of a comprehensive lexicon or resource that captures the semantic relationships between words in the English language. Specifically, a table will be made of the 20 000 most common English words, and for each word in the list, related words that have similar meanings (semantically similar words) and words that make the similar words differ from each other will be found. This table will provide a comprehensive list of words that relate to each other semantically, and it will also indicate how related words differ from their semantically similar words. This lexicon will provide a valuable resource for the NLP and AI research communities, natural language understanding, and many other areas.

To achieve this goal, the thesis will conduct a thorough review of the existing literature on semantic similarity and difference in English, and collect a corpus of text in English to identify the 20 000 most common English words and to identify semantically similar and differentiating words. The methodology will involve developing a program that uses a combination of different libraries and pre-trained models to identify and compare word meanings, including the Natural Language Toolkit (NLTK), WordNet, Wikipedia co-occurring words dataset, and GloVe. The result of this thesis is a table of 20 000 common words with connected similar and differentiating words.

The outcome of the thesis will be validated through a variety of methods, including comparison with existing resources and expert evaluation. The created dataset will be shared with the research community as an open resource for others to use and build upon.

This thesis is written in English and is 33 pages long, including 6 chapters, 1 figure and 3 tables.

Annotatsioon

SEMANTILISE SARNASUSE JA ERISTATAVUSE LEKSIKON: ANALÜÜS 20 000 KÕIGE LEVINUMAST INGLISKEELSEST SÕNAST

Käesoleva lõputöö eesmärk on käsitleda tervikliku leksikoni või ressursi puudumist, mis kajastaks inglise keele sõnade vahelisi semantilisi seoseid. Täpsemalt on töö eesmärk koostada 20 000 kõige levinumast ingliskeelsest sõnast koosnev nimekiri ning leida iga nimekirjas oleva sõna jaoks teised sõnad, millel on sarnane tähendus, ning sõnad, mille poolest sarnased sõnad üksteisest erinevad. Töös loodav tabel annabki põhjaliku loetelu sõnadest, mis on üksteisega sarnase tähendusega, ning lisaks ka sõnadest, mis abistavad sarnaste sõnade üksteisest eristamisel. See leksikon on väärtuslik ressurss NLP ja AI uuringute, loomuliku keele mõistmise ja paljude teiste valdkondade jaoks.

Mainitud eesmärgi saavutamiseks vaadatakse lõputöös põhjalikult läbi olemasolev kirjandus inglise keele semantilise sarnasuse ja erinevuse kohta ning kasutatakse ingliskeelseid tekstikorpuseid, et tuvastada 20 000 kõige levinumat ingliskeelset sõna ja tuvastada semantiliselt sarnased ja ning sarnaseid sõnu eristavad sõnad. Metoodika hõlmab programmi ja andmekogumi väljatöötamist, mis kasutab sõnade tähenduste tuvastamiseks ja võrdlemiseks erinevate eeltreenitud mudelite kombinatsiooni, sealhulgas Natural Language Toolkit (NLTK), WordNet, Wikipedia lähedalpaiknevate sõnade andmekogum ja GloVe.

Lõputöö tulemust valideeritakse erinevate meetodite abil, sealhulgas võrdlemine olemasolevate ressurssidega, eksperthinnan. Leksikoni jagatakse teaduskogukonnale avatud ressursina, mida teised saavad kasutada.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 33 leheküljel, 6 peatükki, 1 joonist, 3 tabelit.

List of abbreviations and terms

NLP	Natural language processing
AI	Artificial intelligence
POS	Part-of-speech
NLTK	Natural Language Toolkit
SYNSET	Set of synonyms
LSA	Latent Semantic Analysis
LDA	Latent Dirichlet Allocation

Table of contents

List of tables

List of figures

1 Introduction

The field of natural language processing (NLP) and artificial intelligence has witnessed significant advancements in recent years, driven by the increasing demand for intelligent systems capable of understanding and interacting with human language. This thesis delves into the concept of semantic similarity between words, a critical aspect of NLP that plays a vital role in various applications such as information retrieval, machine translation, and sentiment analysis, and its potential to improve first-order reasoning in systems like the one presented in "An Experimental Pipeline for Automated Reasoning in Natural Language" [1].

Furthermore, the author combines different approaches to create a dataset of semantically similar words, where similar word pairs have differentiating words attached to them. This is a valuable dataset for resolving contradictory knowledge items in first-order reasoning.

1.1 Overview

Semantic similarity is a fundamental concept in natural language processing (NLP) and plays a significant role in various applications, such as information retrieval, machine translation, and text summarization. The primary goal of semantic similarity measures is to quantify the degree of relatedness between two pieces of text, often sentences or words. Traditional approaches for measuring semantic similarity typically relied on lexical resources such as WordNet, while more recent techniques utilize word embeddings and neural networks. This semantic similarity, as our hypothesis suggests, can help automated reasoning systems deal better with contradictory knowledge items, a concept derived from the aforementioned paper [1].

To address this, the author has developed software that generates a dataset of the 20 000 most common English words and their semantically similar words [2]. Each word pair in this dataset is accompanied by a list of differentiating words, making these similar words distinct from each other [1]. The differentiating words are those that represent the

semantic differences between similar words and can help logic-based systems avoid false assumptions and contradictions during reasoning.

Several studies have contributed to the field of semantic similarity, including a human-inspired method for measuring semantic similarity between sentences [3]. Other research, such as the work by Ezzikouri [4] has introduced new approaches for calculating semantic similarity between words using WordNet and set theory. Additionally, the SensEmbed study [5] aimed to learn sense embeddings for word and relational similarity, and the work by Iqbal et al. focused on measuring semantic similarity in Bengali using word embeddings. Additionally, the author's work also takes inspiration from the experimental pipeline for automated reasoning in natural language [1], providing an avenue for the practical application of the generated dataset.

1.2 Objective

The primary goal of this thesis is to leverage semantic similarity to improve first-order reasoning in automated systems, thus helping them cope better with contradictory knowledge items. To achieve this, the author created a dataset consisting of semantically similar word pairs for the 20 000 most common English words[6], each accompanied by differentiating words [2].

To create this dataset, the author utilized resources like WordNet, ConceptNet, existing word cooccurrence datasets, and word embeddings, and combined the strengths of traditional and modern techniques in natural language processing. The dataset was designed to be both interpretable and accurate in measuring semantic similarity, and its effectiveness was evaluated based on expert evaluation.

The dataset creation process involved the following steps:

- Identifying semantically similar words to the 20 000 common words[6] using WordNet and word embeddings [3]–[5], [7]
- Extracting unique words from sister terms and their explanations in resources like WordNet to identify the words that make similar words different [8].

- Evaluating the dataset based on expert evaluation to ensure its accuracy and effectiveness in measuring semantic similarity and identifying differences between semantically similar words [9].

The resulting dataset can be used to guide automated reasoning systems like the one described in "An Experimental Pipeline for Automated Reasoning in Natural Language" [1] to avoid logical inconsistencies and handle contradictory knowledge better.

1.3 Scope and limitations

This thesis focuses on the following aspects of semantic similarity:

- The exploration and usage of both traditional and modern methods, including lexical resources like WordNet, word embeddings, and neural networks.
- The development of a hybrid model that integrates the most effective features of these methods to achieve higher interpretability and accuracy in measuring semantic similarity.
- The evaluation of the proposed model and real-world applications to assess its effectiveness.

Limitations:

Despite the comprehensive approach taken in this thesis, there are certain limitations to be considered:

The thesis primarily focuses on English language data, and the developed model might not be directly applicable to other languages without modifications or adaptations.

The performance of the proposed model may be influenced by the quality and coverage of the resources used, such as WordNet and the specific word embeddings. These resources may have biases or limitations that could affect the results.

The model's interpretability may still be limited by the complexity of natural language and the inherent difficulties in representing semantic meaning in a computable form.

Due to time and resource constraints, the author may not be able to explore every possible method or technique in depth. The focus will be on those that are most relevant and

promising for the goals of this thesis. While the proposed dataset is expected to improve upon existing methods, it may not be able to eliminate all weaknesses or achieve perfect performance in every scenario. There may still be room for future improvements and refinements.

1.4 Thesis structure

This thesis is organized into several chapters, each serving a specific purpose in addressing the research question. The first chapter provides an introduction, presenting the background, objectives, scope, and limitations of the study. Following the introduction, the second chapter offers a comprehensive literature review that discusses the most relevant methods and techniques for measuring semantic similarity, both traditional and modern, and highlights the strengths and weaknesses of each approach.

The third chapter outlines the methodology employed in this research, detailing the design and development of the created dataset for semantic similarity. This chapter also explains the integration of different techniques and resources, such as WordNet and word embeddings, to create a more accurate and interpretable model.

In the fourth chapter, the implementation and evaluation of the created dataset are presented, including the use of expert evaluation, benchmark datasets, and real-world applications to assess its performance. A comparison of the model's results with those obtained from existing methods is also provided to demonstrate the results that were achieved.

The fifth chapter discusses the findings of the study, highlighting the key contributions of the dataset and addressing any limitations or challenges encountered during the research process. This chapter also provides insights and recommendations for future work in the area of semantic similarity.

Finally, the sixth chapter concludes the thesis by summarizing the main points, reiterating the significance of the research, and emphasizing the potential impact of the proposed model on the field of natural language processing.

2 Background

2.1 Natural language processing

In this section, the author provides an overview of natural language processing (NLP), a subfield of artificial intelligence that focuses on the interaction between computers and human languages. NLP aims to enable computers to understand, interpret, and generate human language in a way that is both meaningful and useful. The author discusses the main components and challenges of NLP, as well as its relevance to the study of semantic similarity.

2.1.1 Components of natural language processing

The key components of NLP include syntactic and semantic analysis, discourse and pragmatics, and language generation. The syntactic analysis focuses on the structure and grammar of sentences, while semantic analysis deals with the meaning of words and phrases in context [7]. Discourse and pragmatics involve the study of how context and speaker intentions influence language interpretation, and language generation refers to the process of producing human-like text based on certain inputs and constraints. These components form the foundation for exploring various techniques and approaches to semantic similarity in subsequent sections.

2.1.2 Challenges in natural language processing

This section delves into the inherent challenges associated with NLP, such as ambiguity, context-dependence, and the vast variability of human languages [Interpretable Semantic Textual Similarity: Finding and explaining differences between sentences]. The author also discusses the limitations of traditional rule-based approaches and the rise of data-driven methods, such as machine learning and deep learning, to tackle these challenges more effectively. Additionally, the section highlights the importance of developing robust and interpretable NLP techniques to address the complexities of human language and improve the performance of various applications.

2.1.3 Relevance of NLP to semantic similarity

NLP is important in studying semantic similarity, as many NLP tasks require the ability to compare and contrast the meanings of words, phrases, or entire texts [8]. By developing

more accurate and interpretable semantic similarity measures, researchers can enhance the performance of various NLP applications, such as machine translation, information retrieval, and text classification, among others [3]. In conclusion, the author emphasizes the role of NLP in understanding and measuring semantic similarity and underscores the need for continued research and development in this area to advance the field.

2.2 Semantic similarity

Analysing semantic similarity is the process of determining how closely related two pieces of text or words are in terms of meaning. Understanding semantic similarity is crucial for many NLP applications, as it enables computers to identify and quantify the relationships between linguistic elements.

2.2.1 Types of semantic similarity measures

Three categories of semantic similarity measures exist: knowledge-based, corpus-based, and hybrid approaches. Knowledge-based methods rely on structured resources such as WordNet, a lexical database that organizes words into hierarchies based on their meanings [4]. Corpus-based techniques, on the other hand, use large collections of text to compute statistical measures of similarity, leveraging methods such as word embeddings and co-occurrence matrices [9].

Hybrid approaches combine elements of both knowledge-based and corpus-based methods to achieve more accurate and robust results. These methods often incorporate additional linguistic features or use machine learning techniques to learn better similarity representations from data [5].

2.2.2 Challenges in measuring semantic similarity

Measuring semantic similarity is not an easy task due to several challenges. The author discusses these challenges, which include the inherent ambiguity and context-dependence of natural language, as well as the difficulty of defining a universally applicable measure of similarity [7]. Moreover, the author acknowledges the trade-offs between simplicity and accuracy when developing similarity measures, as simpler methods may not capture the nuances of language, while more complex approaches might be computationally expensive or difficult to interpret [10].

These challenges highlight the importance of continued research and development in the field of semantic similarity, as well as the need for novel methods that can strike a balance between accuracy, interpretability, and computational efficiency.

2.2.3 Applications of semantic similarity

There are various applications of semantic similarity in NLP. These applications include tasks such as information retrieval, machine translation, text classification, and sentiment analysis, among others [8]

In addition to these NLP tasks, semantic similarity can also play a significant role in improving first-order reasoning in automated systems. By comprehending the semantic relationship between words and their distinct differences, these systems can avoid logical inconsistencies and handle contradictory knowledge more efficiently.

For instance, a typical problem in first-order reasoning is the generation of false logical inferences, such as "king is similar to queen, king is male and male people might have a beard, therefore queen might have a beard." The dataset created in this thesis, which includes semantically similar words and words that make them different, can help prevent such fallacies. The differentiating words can provide an additional layer of information that can help these systems distinguish the subtle semantic differences between similar words.

The development of the dataset proposed in this thesis, which includes semantically similar words and words that make them different, is an example of how research in semantic similarity can lead to new resources and insights for the field and improve the performance of automated reasoning systems [3].

In summary, the study of semantic similarity is a vital aspect of NLP, as it enables computers to better understand and process human language. It also holds much value for improving first-order reasoning in automated systems. Despite the challenges and complexities involved, continued research in this area holds great potential for improving a wide range of NLP tasks and applications, and the performance of automated reasoning systems.

2.3 Word embeddings

Word embeddings are a popular technique in NLP that involve mapping words or phrases to continuous vectors of fixed dimensions [11]. These vectors are designed to capture the semantic meaning and relationships between words, allowing for more effective representation and manipulation of textual data. Word2Vec and GloVe are two widely-used algorithms for generating word embeddings, each with its own approach to capturing context and semantics.

Word2Vec leverages the surrounding words in a given text to learn vector representations, following the principle that words with similar contexts have similar meanings [12]. GloVe, on the other hand, focuses on the co-occurrence statistics of words in a corpus to learn their vector representations, providing a more global understanding of word relationships [13].

Since word vectors are still vectors and can be used to make calculations some functions and calculators allow you to make computations such as “king – man + woman”. One might arrive at an answer such as “queen”, but what is required is for it to return a big list of words. Such methods were also tested in this thesis, though it doesn’t apply to our use case, since such methods give us a vector to which one could find words that have similar scores (cosine similarity for example) [14]. An example of what we’re looking for is a program or function that by inputting “king – queen” would output “man, woman, male, female, strong, weak” etc. The author couldn’t find any promising arguments or examples that would promise or hint at the chance of giving promising results.

The use of word embeddings has been shown to improve the performance of tasks such as text classification, sentiment analysis, and machine translation [15]. However, one limitation of word embeddings is their inability to capture multiple meanings of a single word, known as polysemy. This has led to the development of more advanced techniques like sense embeddings, which represent distinct senses of words separately, thereby addressing the issue of polysemy [5].

Overall, word embeddings serve as a critical foundation for many semantic similarity approaches and contribute to the ongoing advancements in the field of NLP. The dataset developed in this thesis, which includes semantically similar words and words that make them different, leverages word embeddings, specifically the GloVe 300d model, to

measure the similarity and difference between words. According to expert evaluation, this model was found to be the most suitable and accurate among the various models tested.

2.4 Existing approaches and tools

Over the years, several approaches and tools have been developed to address the problem of computing semantic similarity between words or sentences [16]. These methods can be broadly classified into three categories: knowledge-based, corpus-based, and hybrid techniques.

Knowledge-based methods rely on structured knowledge resources such as WordNet, a lexical database that organizes words into hierarchies based on their meanings [17]. Approaches using WordNet often measure semantic similarity by calculating the shortest path between two words in the hierarchy, the depth of their shared hypernym, or by employing information content measures. While knowledge-based techniques provide interpretable results, they may suffer from coverage limitations, as not all words or concepts are present in the knowledge resources.

Corpus-based methods, in contrast, use the distributional hypothesis, which suggests that words found in similar situations usually have similar meanings [18]. These methods utilize large text corpora to learn statistical patterns and relationships between words. Word embeddings, as discussed in the previous section, are a popular example of corpus-based approaches. Latent Semantic Analysis (LSA) and topic modeling techniques, such as Latent Dirichlet Allocation (LDA), are other examples that use the co-occurrence of words to extract latent semantic structures from text data [19], [20]. Despite their ability to capture semantic information from large-scale data, corpus-based methods can be computationally intensive and may struggle with rare words or phrases.

Hybrid techniques combine the strengths of both knowledge-based and corpus-based approaches to overcome their limitations. These methods typically integrate information from structured knowledge resources like WordNet with statistical patterns derived from text corpora. For example, some hybrid approaches enrich word embeddings with information from WordNet, allowing for a more comprehensive understanding of word relationships and semantics. [15]

Numerous tools have been developed to implement these approaches, ranging from open-source libraries like Gensim, SpaCy, and NLTK [21]–[23] to commercial solutions like IBM Watson and Google's Natural Language API [24], [25]. These tools enable researchers and practitioners to easily apply advanced semantic similarity techniques to various NLP tasks, driving innovation and progress in the field.

In this thesis, the dataset created for semantically similar words and words that make them different leverages both knowledge-based and corpus-based approaches, including the use of WordNet and word embeddings like GloVe. By combining these methods, the dataset provides a comprehensive resource for exploring the differences between semantically similar words.

3 Methodology

This section outlines the methodology adopted in this thesis to address the problem of computing semantic similarity between words. The approach consists of several steps, including data collection, preprocessing, feature extraction, and evaluation.

3.1 Exploration of traditional and modern methods

There are traditional and modern methods in NLP. Traditional methods are more simplistic and use techniques that have been in use for a longer time than modern methods, which have appeared with machine learning techniques such as deep learning.

3.1.1 Traditional methods

Traditional methods in NLP primarily revolve around knowledge-based techniques that rely on pre-existing linguistic resources, such as dictionaries and thesauri, to establish semantic relationships between words.

- WordNet: WordNet is a large lexical database of English words, which organizes words into synonym sets, or synsets. It captures semantic relationships between words, such as synonymy, antonymy, hypernymy, and hyponymy. For example, in WordNet, the word "car" is connected to the hypernym "automobile" and the hyponym "sedan."

- Distributional semantics: This approach is based on the idea that words that occur in similar contexts tend to have similar meanings. It involves creating co-occurrence matrices of words in a given corpus and calculating similarity measures, such as cosine similarity or Jaccard similarity. For instance, the words "dog" and "cat" might be considered similar because they often appear in similar contexts.

3.1.2 Modern methods

Modern methods in NLP are predominantly data-driven and leverage machine learning techniques, particularly deep learning, to learn semantic representations from large text corpora.

- Word embeddings: Word embeddings, such as Word2Vec, GloVe, and FastText, are continuous vector representations of words that capture semantic and syntactic relationships between them. These models are trained on large corpora and can be used to compute similarity scores between words. For example, the Word2Vec embedding of "king" and "queen" would have a high similarity score, reflecting their semantic relationship. It is also possible to add and subtract word vectors to and from each other, where the result is a word vector to which one can find similar words [14].

- Transformer-based models: These models, such as BERT, and RoBERTa, have achieved state-of-the-art performance in various NLP tasks by learning contextualized word representations. Unlike traditional word embeddings, these models can capture context-dependent semantic relationships, which can be helpful in disambiguating words with multiple meanings. For instance, BERT can differentiate between the word "bank" in the context of a financial institution and a riverbank.

In conclusion, both traditional and modern methods have their advantages and limitations when it comes to capturing and representing semantic relationships between words. Traditional methods can provide explicit knowledge about word relationships, but may struggle with ambiguous or context-dependent meanings. In contrast, modern methods excel at learning implicit relationships from large amounts of data but may require significant computational resources and are often seen as "black boxes." By exploring the combination of these methods, we aim to develop a more comprehensive understanding of semantic relationships and improve the performance of NLP tasks.

3.2 Hybrid approach

The hybrid approach combines the strengths of both traditional and modern methods to create a more robust and comprehensive representation of semantic relationships between words. This approach aims to harness the explicit knowledge provided by traditional methods and the implicit, data-driven knowledge learned by modern methods.

3.2.1 Motivation

The motivation behind the hybrid approach is to address the limitations of traditional and modern methods in capturing semantic relationships. Traditional methods, while providing explicit knowledge, may lack the ability to adapt to new or evolving language usage. On the other hand, modern methods, though effective in learning implicit relationships, can be computationally expensive and may not always provide interpretable results.

By integrating both approaches, we can leverage the advantages of each while mitigating their weaknesses. This results in a more comprehensive and adaptable system that can better capture the nuances and complexities of semantic relationships in natural language.

3.2.2 Methodology of the hybrid approach

To implement a hybrid approach, we follow these steps:

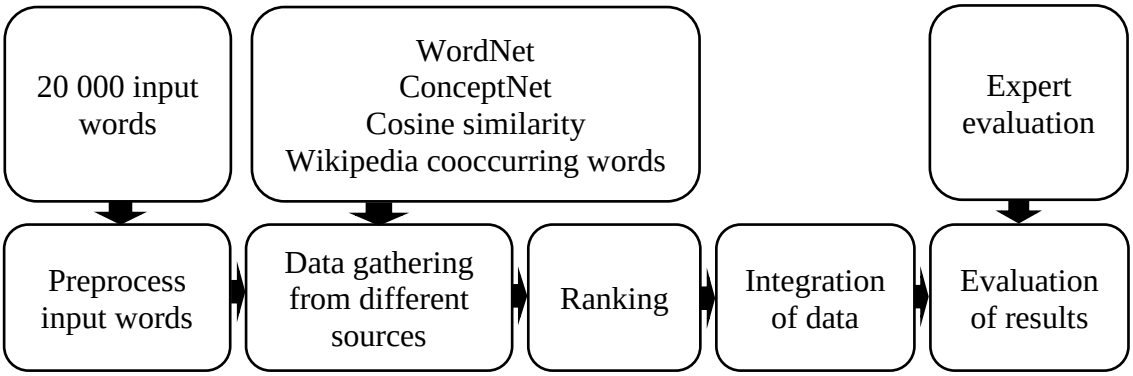

Figure 1. The steps and methodology of the hybrid approach.

Preprocessing: We first clean and preprocess the text data to remove any inconsistencies, such as special characters, numbers, or common words (stopwords). We also turn each word into its base form, a process known as lemmatization. This makes it easier to process, compare, and analyse the words later on.

Data gathering: After preprocessing, we gather words from various sources. This includes cooccurring words from Wikipedia [26], descriptive words from WordNet [17] sister terms, descriptive words from cosine similarity, and related terms from ConceptNet [27].

Ranking: Once we have gathered the data, we analyse each word. If a word appears multiple times across the different sources, it's likely important, so we move it to the front

of the differentiating word list. Each word also has weights attached to them, providing additional information about its importance.

Integration: After ranking and weighting, we bring together all the information we've gathered into one unified dataset. This involves considering the word's rank and weight from each source.

Evaluation: Lastly, we evaluate how well the hybrid approach works using expert evaluation.

3.3 Data collection

The first step in our methodology is to collect a suitable dataset. We start with the 20 000 most common English words, which we get from the internet [6]. To find words that are similar to each of these, we use a combination of methods including the cosine similarity function, WordNet, ConceptNet, and cooccurring words from Wikipedia [26].

Next, we find similar words for these similar words. We do this again using the same functions, but this time, we only pick out verbs and adjectives from the list, because by using trial and error we found that adjectives and verbs describe the differences between similar words the best, based on expert evaluation.

By using these varied sources, we ensure that our approach applies to different contexts. For this, the dataset needs to be diverse and should represent the problem domain well.

3.4 Data preprocessing

After collecting the data, preprocessing steps are performed to clean and ready the text for further analysis. This might involve tokenization, where the text is separated into individual words or tokens, and normalization, such as converting all text to lowercase and removing punctuation marks. Stop words, which are common words with little semantic meaning (e.g., "the", "and", "is"), are added to the list of "stop words" to be ignored when looking for differentiating words, since they do not seem to contribute to finding words that describe the differences between two words. Furthermore, stemming or lemmatization techniques may be used to simplify words to their base forms, facilitating better comparison between words with similar meanings but different forms.

Some information might get lost here, since initial words without context lack the information that is needed. For example, the word “can” might mean “be able to” or it might mean “a container for storing material”.

3.5 Identifying sister terms

In this step, the goal is to identify sister terms for the input words. Sister terms are words that share a common hypernym or broader category in a lexical resource like WordNet. Identifying sister terms can provide additional context and help in understanding the semantic relationships between words.

An example of such sister term is “prince” and “princess” because they share a common hypernym. To achieve this, the NLTK library is utilized to access WordNet, extract synsets for the input words, and obtain their hypernyms. Once the hypernyms are retrieved, the hyponyms of these hypernyms are gathered, resulting in a list of sister terms for each input word.

3.6 Retrieving related terms from ConceptNet

In this step, we aim to retrieve related terms from ConceptNet, a semantic network representing words and phrases and their associated ideas and meanings. ConceptNet is a useful resource in our methodology as it helps to enhance the semantic relationships between words by providing a wide range of related terms.

For instance, if we take the word “tree”, ConceptNet can provide related terms like “plant”, “wood”, “nature”, and “forest”. These related terms not only expand our understanding of the word “tree” but also provide additional context that can be crucial in discerning semantic relationships.

In addition to providing a broader context, ConceptNet can also help in identifying more specific relationships. For instance, for the word “run”, related terms might include “jog”, “sprint”, or “dash”, each conveying a slightly different nuance of the action.

3.7 Extracting cooccurring words from the Wikipedia corpus

The next phase involves extracting cooccurring words from a pre-existing Wikipedia dataset [26] that was created and handed to me by one of the supervisors of this thesis. This dataset, which comprises words that frequently appear together in Wikipedia articles, serves as a valuable resource for our methodology.

Cooccurring words can offer valuable insights into how words are commonly used together in natural language, providing a more realistic and context-rich understanding of their relationships. For instance, in the case of the word “rain”, frequently cooccurring words might include “cloud”, “weather”, or “umbrella”. These words not only give us a better understanding of the context in which “rain” is often used but also help to differentiate it from semantically similar words.

3.8 Extracting descriptive words

After identifying the sister terms, the next step is to extract descriptive words from their definitions in WordNet. Descriptive words are adjectives or verbs that characterize the sister terms, providing valuable information about their meanings and the context in which they are used. For example, descriptive words for the word “king” might be “strong”, “rule” and “conquer”.

To extract descriptive words, the NLTK library is employed once again to tokenize and part-of-speech (POS) tag the definitions of sister terms. The tokens with POS tags corresponding to adjectives (“JJ”) and verbs (“VB”) are then collected and added to a list of descriptive words. Different weights were tested for adjectives and verbs, resulting in a mix of 90% adjectives and 10% verbs in the final list of descriptive words.

3.9 Semantic similarity and difference calculation

With the descriptive words obtained, the semantic similarity and difference between the input words can now be calculated. To do this, word embeddings, such as GloVe or Word2Vec, are employed to represent the descriptive words as high-dimensional vectors. The similarity and difference between these vectors can then be measured using cosine similarity, which ranges from 0 (completely different) to 1 (identical).

For differentiating words, adjectives and verbs were extracted from descriptions of words. For example, from the description of “king” as a "male monarch, Rex (a male sovereign; ruler of a kingdom)", the word “male” would be extracted and added to the list of descriptive words for “king”. The most descriptive verbs and adjectives were found by using cosine similarity, descriptive words from sister terms from WordNet, related terms from ConceptNet, and co-occurring words from the Wikipedia dataset [26]. When comparing similar words, such as “king” and “queen”, a symmetric difference was taken from their lists of descriptive words. This means that shared descriptive words, such as “powerful”, were discarded while differentiating words like “male” and “female” were considered. The top words of the combined list of differentiating words were selected using the count of words from different sources. If, for example, a ConceptNet and sister terms’ descriptive words both returned “male” as a descriptive word for the word “king”, then the count was increased (to 2 and so forth) and it moved up to the top of differentiating words. After that, the list was mixed, alternating between all of the sources if the counts of words were equal. When alternating between sources for creating the mixed list, the words were still ordered by weights within each source, as that was the next best indicator of a descriptive word, since it carried at least some extra information. The list continued in this manner until no more words were available from any of the source lists

By calculating the semantic similarity and difference using the combined list of descriptive words, the proposed methodology enables a more accurate and nuanced understanding of the relationships between words. This approach can be applied to various contexts and problem domains, offering valuable insights into the semantic structure of language.

3.9.1 Tested models

We evaluated several pre-trained word embedding models. To determine the best model for computing semantic similarity and difference, we assessed each model based on a set of key criteria, each rated on a scale from 1 to 5.

Performance: This criterion evaluates the accuracy and effectiveness of a model. We consider how well the model can identify words that are semantically similar or different. For instance, if the model can correctly identify that “dog” and “puppy” are similar but

“dog” and “book” are not, it would score highly in this category. Correctly identifying similar words means assigning a high similarity score to them on a scale from 0 to 1

Computational requirements: We also consider the resources needed to run the model. Models that require fewer resources—such as less memory or processing power—are favoured. We aim for efficiency, so a model that delivers accurate results with minimal resource usage would score highly.

Output words’ usability: This criterion assesses how easily the output words can be used in the required context. If the model generates words that fit seamlessly into various contexts and can be easily understood, it would earn a high score. For instance, if the model suggests “puppy” as a similar word to “dog”, it indicates high usability.

Suitability: Lastly, we assess how well the model suits the overall goal of the project. This includes factors such as the model's ability to handle the size of our dataset and its compatibility with other tools we're using. If it is difficult to put to use or needs a lot of time and effort to get to work or is not possible to use with the author’s skillset, then a low score will be assigned.

Table 1. Comparison of pre-trained language models. The models were ranked on a scale of 1..5 with 5 being the highest score.

Model name	**Performance**	**Computational requirements**	**Output words' usability**	**Suitability for the end goal**
GloVe	4	5	4	5
Numberbatch	3	3	2	3
Brown	3	4	4	4
FastText	3	3	3	3
Google News	3	2	4	3
BART	3	1	3	3
RoBERTa	4	1	3	3

- GloVe (Global Vectors for Word Representation) with different dimensions: 100d, 200d and 300d. [13]
    - Pros: fast loading time, simple words, different dimensions to improve loading time for testing
    - Cons: word-noise (for example roman numerals like "III", "II" etc are considered valid words)
- Numberbatch 19.08, a semantic vector model derived from multiple sources, including GloVe and Word2Vec. [28]
    - Pros: combined from multiple sources, very comprehensive
    - Cons: cluttered with phrases (instead of words), slow loading time, excess of information

- Brown, a corpus-based distributional model that captures word co-occurrence patterns.
    - Pros: captures word relationships, moderate loading time
    - Cons: limited vocabulary, less accurate for complex semantic relationships
- FastText, a library for learning word representations that considers subword information. [29]
    - Pros: captures subword information, suitable for rare words
    - Cons: slower loading time, not as accurate for semantic similarity and difference tasks
- Google News Model, a pre-trained Word2Vec model based on the Google News dataset. [30]
    - Pros: a large vocabulary, updated with recent news data
    - Cons: slow loading time, focuses on news-related words
- BART (Bidirectional and Auto-Regressive Transformers), a pre-trained sequence-to-sequence model that can generate natural language text. [31]
    - Pros: advanced natural language understanding suitable for generating text
    - Cons: high computational requirement, not designed for semantic similarity and difference tasks, requires transforming the model
- RoBERTa (Robustly optimized BERT), a pre-trained language model based on the BERT architecture. [32]
    - Pros: advanced natural language understanding, captures complex relationships
    - Cons: high computational requirement, not specifically designed for semantic similarity and difference tasks, requires transforming the model

The performance of these models was compared to identify the most effective model for the specific task of calculating both semantic similarity and difference in this thesis.

After a thorough evaluation and comparison of the performance of these models, the GloVe 300d model emerged as the most suitable and accurate for computing semantic similarity and difference in this thesis. Human evaluation of the results generated by the different models showed that the GloVe 300d model provided the closest match to expert evaluation of similarity and difference. Consequently, this model was chosen for further analysis and implementation in the proposed system.

3.10 Results

The output [2] is generated in a JSON format, presenting the input words alongside their related words, similarity scores, and another table where there are similar words next to each other and also a list of descriptive words that contributed to the difference. The tables are organized in such a way that it is easy to understand and interpret the semantic relationships between the input words and their related words. The tables are merged into one JSON file for ease of access and usability. All of the differentiating words are in the order of importance – if different sources returned male as a descriptive word of “king”, then each time it was counted, and if the similar word didn’t have that descriptive word (which would cancel it out), then it made that descriptive word more important.

Below are the tables extracted from the JSON file, illustrating both intuitive and unintuitive results – based on expert evaluation. The tables present the input word, a similar word, the similarity score, and the list of related words that contribute to the difference between the input word and the related word (differentiating word). For the sake of readability, the list of differentiating words was truncated in the following two tables.

For the results to be suitable it is necessary to have differentiating words exist in the “differentiating words” column. Examples of expected and intuitive words have been highlighted with bold text. While there are words that are not perfectly intuitive or seem excessive, it does not subtract from the value of the differentiating words list. The goal was to find words that make two words differ from each other.

Before we go any further it must be said, that with tools available to the author, it is quite difficult to arrive at differentiating words between input words. As of so far, no articles, references, examples nor even arguments were found that would offer solutions to our goal.

In Table 2, the results showcase the effectiveness of the methodology in capturing the semantic similarity and differences between words. For example, the words "king" and "queen" have a similarity score of 0.634, and the list of differentiating words includes "male" and ". Also, "queen" and "princess" are great examples of good results, as there are words like "old", "small", "single" in the list of differentiating words, which are expected and more importantly – wished for. Words that are not highlighted might apply for either of the words – input for or similar word or are just of low value for our main use case, which is to find distinct differentiating words.

Table 2. Excerpts from expected and intuitive results.

Input word	**Similar word**	**Similarity score**	**Differentiating words**
king	queen	0.634	**male**, british, marry, ruler, immediate, golden, knight, monarch, legal, succeed, youngest, **man**, single, swedish, grand, **female**..
man	person	0.644	old, **male**, boy, dead, particular, little, adult, live, social, alive, **gender**, amaze, significant, turn, **guy**, black, such, look, human, rich..
queen	princess	0.636	golden, unoccupied, name, king, **old**, move, famous, castle, **small**, **little**, powerful, knight, rule, homecoming, **single**, royalty, white..
machine	automatic	0.526	mechanical, such, portable, work, make, sexual, **electrical**, glaze, **electric**, political, conventional, filter, sew, social, **power**, machine-controlled..
banana	mango	0.580	**yellow**, delicious, **soft**, annual, important, mexican, herb, eastern, slice, import, purple, leave, brandied, fruit..
grass	lawn	0.543	small, form, soft, **park**, nest, dried, deciduous, **field**, native, animal, flat, ground, artificial, prevent, hot, plant, natural, blanket-like, shade..
car	truck	0.735	**load**, **speed**, city, electric, same, crash, din, carry, seat, new, **haul**, automobile, buy, explode..

However, as illustrated in Table 3, there are examples, where we couldn't find any differentiating words. Or rather they might be differentiating and are relevant, just not in an intuitive way.

A closer examination of the results in both tables reveals that there are numerous instances where the word pairs or differentiating words do not seem to make much sense from a human perspective. This might be attributed to limitations of the underlying methodology, which may not account for certain nuances in language or semantic relationships between words.

However, it is essential to consider that even the unexpected outcomes from these tables might be helpful and suitable for use in semantic tasks, where concepts in our minds do not relate that closely to the needed inputs for NLP programs or other similar software that improve their performance. This is because NLP and machine learning models often rely on patterns and statistical relationships within the data, which might not always align with human intuition or understanding. In some cases, these unintuitive relationships could provide valuable information for the models to learn from and adapt their behaviour accordingly [33].

As researchers continue to explore and develop new techniques to improve the interpretability and accuracy of semantic similarity measures, it is crucial to recognize the potential value of seemingly nonsensical or unintuitive results. By acknowledging the limitations of current methods and considering the potential usefulness of these unexpected outcomes, we can continue to push the boundaries of NLP research and develop more effective and efficient tools for processing and understanding human language.

Table 3. Example of unsuccessful results.

Input word	**Similar word**	**Similarity score**	**Differentiating words**
piano	violin	0.865	other, upright, compose, music, imitate, grand, perform, concert, click, international..
piano	clarinet	0.808	singing, vocal, acoustic, playing, musical, classical
parent	sibling	0.464	unmarried, haploid, private, organism, settle, free, acquire, separate, diploid, buy, come, require, mutual, find, apparent, own, autistic, dominant..
motorcycle	car	0.602	drive, sled, garage, rid, skateboard, city, automotive, wheel, vehicle, british, speed, crash, japanese, same, seat, indian, carry, automobile, **large**..
vegetarian	vegan	0.726	delicious, animal, non-vegetarian, vegetable, healthy, barbecue, indian, nonalcoholic, dumpling, non-alcoholic, many, sedentary, chinese, plant-based

4 Implementation

To run the code in the GitLab repository [2] some system requirements should be met to make sure that the code will work. It is recommended to use Linux since the whole software was tried and tested on it.

4.1 System requirements

To successfully implement the proposed methodology, the following system requirements should be met:

- A computer with a modern operating system capable of running Python 3.
- Sufficient memory (RAM) to process and store the word embeddings and WordNet data, with at least 2 GB of available memory recommended.
- Python 3 is installed, along with the necessary libraries and packages mentioned below.

4.2 Tools and libraries

4.2.1 NLTK

The Natural Language Toolkit (NLTK) is a Python library that provides a comprehensive suite of tools for working with human language data. In this project, NLTK is used to access WordNet, tokenize and POS-tag the definitions of sister terms, and extract descriptive words.

4.2.2 PrettyTable

PrettyTable is a Python library that enables the creation of simple ASCII tables. It is used to display the output in a tabular format that is easy to understand and interpret.

4.2.3 JSON

The JSON library is a built-in Python module that allows for encoding and decoding JSON data. It is used to load and process JSON files containing word embeddings and WordNet data.

4.2.4 Resource

The resource library is another built-in Python module that is used to manage system resources, such as memory limits. In this project, the resource library is employed to set a memory limit for processing and storing word embeddings and WordNet data. No installation is required, as it is included in the standard Python distribution.

4.3 Code explanation

In this section, the author provides a brief overview of the main components of the implemented code and explains their purpose and functionality.

4.3.1 NLTK functions

The code utilizes the Natural Language Toolkit (NLTK) library to perform various natural language processing tasks. These tasks include tokenization, part-of-speech tagging, and accessing the WordNet lexical database. The functions word_tokenize() and pos_tag() are used for tokenization and part-of-speech tagging, respectively. The wn.synsets() function is employed to retrieve synsets from the WordNet database.

4.3.2 Finding semantically similar words

The list of semantically similar words was created by using GloVe model with genism library in python to find words, that have the highest cosine similarity score. A high cosine similarity score means that the words appear near each other in the vector space of the model, which in turn implies a similar meaning. The author tested various models and eventually chose the GloVe 300d model due to its superior performance based on expert evaluation.

4.3.3 Extracting descriptive words

Descriptive words from sister terms

The get_sister_terms() function is responsible for identifying sister terms of a given input word. This is achieved by first obtaining the synsets of the input word using the wn.synsets() function, which is all built on the WordNet dataset. The get_type_of_words_from_sister_term_descriptions() function combines the functionality of get_sister_terms() and get_adjectives(). It accepts an input word, a list of words to disregard (words that are too unique to be of any value in this context), and types of words

to be extracted from the sister terms' descriptions, such as verbs and adjectives. It returns a list of descriptive words from the sister terms that match the specified word types.

Descriptive words from the Wikipedia corpus

The author got access to a dataset [26] of co-occurring words that were extracted from Wikipedia. It was compiled by one of the supervisors of this thesis – T. Tammet. From this corpus, a list of cooccurring words was extracted with weights of how often they appeared near each other.

Related words from ConceptNet

ConceptNet has a dataset that is organized in a way that it is possible to retrieve a list of words that are related to the input word. This was used through an API to gather words that were related to each other. These also had weights attached to them.

Similar words from GloVe pre-trained model

By again using the cosine similarity function most_similar() a list of most similar words was found. From there only adjectives and verbs were extracted as it was found that they are most descriptive of the input words.

The words from each source were lemmatized to make them more comparable and remove duplicates.

After combining the retrieved lists of descriptive words they were combined by keeping the order of weights within each source and counting the words that appeared from different sources. If a word appeared more than once, its count went up and made it move up in the importance of differentiating words. If words had the same counts, then they were just alternated between all of the sources because the weights that came with descriptive words from different sources were not comparable.

4.3.4 Displaying the results

The output is generated using the PrettyTable library, which creates a formatted table for displaying the results. The table includes columns for the input word, related word, similarity score, and differentiating words. The table's formatting options are set, such as maximum column widths and horizontal lines, to improve readability.

The code iterates through the user's input words and retrieves the related words, scores, and descriptive words from the JSON data file. These values are added as rows in the PrettyTable output. The final table is displayed to the user, showcasing the semantic similarity results.

5 Results and discussion

In this section, the author presents the results obtained from the implemented system and discusses their implications and relevance to the study's objectives.

5.1 Results

The implemented system successfully identified the semantic similarity between the input word and its sister terms using the GloVe 300d model. The results were presented in a clear and concise tabular format, allowing the user to quickly identify the data by running a Python script which helps to find the sought-after word’s information. The table displays the input word, related word, similarity score, and differentiating words for each comparison.

5.2 Model Comparison

The author tested several word embedding models, including GloVe with varying dimensions (100d, 200d, 300d), Numberbatch 19.08, Brown, FastText, Google News model, BART, and RoBERTa. The GloVe 300d model was found to be the most suitable and accurate based on expert evaluation, providing a balance between performance, computational requirements, usability of output words, and suitability for the end goal.

5.3 Discussion

In this section, we discuss the implications of our findings, the limitations of our methodology, and potential future research directions.

5.3.1 Implications of findings

The results of our semantic similarity and difference, as presented in the tables, indicate that our methodology is capable of identifying semantically related words with a comparatively high degree of success. In some cases, the method produces meaningful and relevant results that align with human intuition. This suggests that our approach has the potential to be used in NLP tasks, such as text classification, sentiment analysis,

information retrieval, and resolving contradictory knowledge items in first-order reasoning.

However, the presence of unintuitive or seemingly nonsensical word pairs and related differentiating words in the results also highlights the limitations of our methodology. These instances might be attributed to inherent biases in the data or shortcomings in the algorithms used to calculate semantic similarity. While some of these unexpected outcomes may still provide valuable input for NLP models, further research is needed to understand the circumstances under which these results are useful or detrimental to model performance.

5.3.2 Limitations of methodology

Our methodology is not without its limitations. First, the choice of the dataset and pre-processing techniques may have introduced biases that affect the results. Moreover, the similarity measure we used may not capture all nuances of semantic relationships between words, leading to unexpected outcomes. Lastly, our method relies on a single metric to assess similarity, which may not provide a comprehensive representation of the true semantic relationship between words.

5.4 Future research directions

Based on our findings and the limitations discussed, several directions for future research emerge:

- Exploration of alternative similarity measures: By comparing the performance of different similarity measures, researchers can identify more effective approaches for capturing semantic relationships between words. This also applies to semantic differences as the general principle of finding differences is comparing similarities and applying symmetric subtraction on the descriptive words of similar words.

- Evaluation of different datasets and pre-processing techniques: Investigating the impact of various datasets and pre-processing methods on semantic similarity results can provide valuable insights into potential biases and sources of error.

- Development of hybrid similarity and differentiation measures: Combining multiple similarity metrics may lead to more accurate and robust representations

of semantic relationships. The software created in this thesis allows for easy testing of different datasets and adding new sources of data and methods.

- Investigation of the utility of unexpected results: Further research is needed to understand the role of seemingly nonsensical or unintuitive outcomes in NLP tasks and whether they can improve model performance under certain conditions.

- Incorporating context awareness: Developing methods that account for context when calculating semantic similarity and differentiation can potentially improve the accuracy and relevance of the results.

In conclusion, our study demonstrates both the potential and limitations of our methodology for measuring semantic similarity and difference. By addressing these limitations and exploring future research directions, we can continue to refine our understanding of semantic relationships and contribute to advancements in the field of NLP.

6 Summary

In this thesis, we have presented an approach to aggregating semantic information from multiple sources, including WordNet, ConceptNet, word cooccurrence dataset, pre-trained models, and NLTK library, to generate two tables of information: one for semantically similar words and one for differentiating words. Our objective was to leverage the strengths of each source to create a more comprehensive representation of semantic relationships, which can be used to enhance the performance of NLP tasks such as resolving contradictory knowledge items in first-order reasoning.

6.1 Key findings and contributions

Our methodology successfully produced tables containing semantically similar and differentiating words, demonstrating the feasibility of combining multiple sources of semantic information. The tables provided meaningful and relevant results in many instances, showcasing the potential of our approach for use in various NLP applications. However, we also identified cases where the generated results were unintuitive or seemed nonsensical, indicating limitations in our methodology.

This thesis contributes to the field of NLP in several ways:

- Development of a novel approach to aggregating semantic information from diverse sources, which can be used to enrich the understanding of semantic relationships between words.
- Presentation of tables containing semantically similar and differentiating words, offering a valuable resource for NLP researchers and practitioners.
- Analysis of the strengths and weaknesses of our methodology, providing insights for future research and improvements.

6.2 Future work

Based on our findings and the limitations of our methodology, we have outlined several future research directions, including exploration of alternative similarity measures, evaluation of different datasets and pre-processing techniques, development of hybrid similarity measures, investigation of the utility of unexpected results, and incorporation of context-awareness in semantic similarity calculations.

In summary, this thesis has demonstrated the potential of aggregating semantic information from multiple sources to better understand and represent semantic relationships between words. By addressing the limitations of our current approach and pursuing the suggested future research directions, we can continue to refine our methodology and contribute to the ongoing advancement of NLP.

References

[1] T. Tammet, P. Järv, M. Verrev, and D. Draheim, ‘An Experimental Pipeline for Automated Reasoning in Natural Language’, *accepted to CADE 2023*, Tallinn University of Technology, Tallinn, Estonia.

[2] R. Vellemaa, ‘Semantic similarity’, *GitLab*, May 05, 2023. [Online]. Available: https://gitlab.cs.ttu.ee/Reimo.Vellemaa/semantic-similarity/-/tree/main [internal TalTech] https://github.com/wargunnerguy/Semantic-similarity-MSc-thesis [public]. [Accessed: May 07, 2023]

[3] J. I. Serrano, M. D. del Castillo, J. Oliva, and R. Raya, ‘Human-inspired semantic similarity between sentences’, *Biol. Inspired Cogn. Archit.*, vol. 12, pp. 121–133, Apr. 2015, doi: 10.1016/j.bica.2015.04.007.

[4] H. Ezzikouri, Y. Madani, M. Erritali, and M. Oukessou, ‘A New Approach for Calculating Semantic Similarity between Words Using WordNet and Set Theory’, *Procedia Comput. Sci.*, vol. 151, pp. 1261–1265, 2019, doi: 10.1016/j.procs.2019.04.182.

[5] I. Iacobacci, M. T. Pilehvar, and R. Navigli, ‘SensEmbed: Learning Sense Embeddings for Word and Relational Similarity’, in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Beijing, China, 2015, pp. 95–105, doi: 10.3115/v1/P15-1010 [Online]. Available: http://aclweb.org/anthology/P15-1010. [Accessed: May 06, 2023]

[6] J. Kaufman, ‘List of the 20,000 most common English words in order of frequency’, May 11, 2023. [Online]. Available: https://github.com/first20hours/google-10000-english. [Accessed: May 12, 2023]

[7] G. Zhu and C. A. Iglesias, ‘Computing Semantic Similarity of Concepts in Knowledge Graphs’, *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 1, pp. 72–85, Jan. 2017, doi: 10.1109/TKDE.2016.2610428.

[8] G. Recski, E. Iklódi, K. Pajkossy, and A. Kornai, ‘Measuring Semantic Similarity of Words Using Concept Networks’, in *Proceedings of the 1st Workshop on Representation Learning for NLP*, Berlin, Germany, 2016, pp. 193–200, doi: 10.18653/v1/W16-1622 [Online]. Available: http://aclweb.org/anthology/W16-1622. [Accessed: May 06, 2023]

[9] Md. A. Iqbal, O. Sharif, M. M. Hoque, and I. H. Sarker, ‘Word Embedding based Textual Semantic Similarity Measure in Bengali’, *Procedia Comput. Sci.*, vol. 193, pp. 92–101, 2021, doi: 10.1016/j.procs.2021.10.010.

[10] I. Lopez-Gazpio, M. Maritxalar, A. Gonzalez-Agirre, G. Rigau, L. Uria, and E. Agirre, ‘Interpretable semantic textual similarity: Finding and explaining differences between sentences’, *Knowl.-Based Syst.*, vol. 119, pp. 186–199, Mar. 2017, doi: 10.1016/j.knosys.2016.12.013.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, ‘Efficient Estimation of Word Representations in Vector Space’. arXiv, Sep. 06, 2013 [Online]. Available: http://arxiv.org/abs/1301.3781. [Accessed: May 07, 2023]

[12] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, ‘Distributed Representations of Words and Phrases and their Compositionality’. arXiv, Oct. 16,

2013 [Online]. Available: http://arxiv.org/abs/1310.4546. [Accessed: May 07, 2023]
[13] J. Pennington, R. Socher, and C. Manning, 'Glove: Global Vectors for Word Representation', in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 1532–1543, doi: 10.3115/v1/D14-1162 [Online]. Available: http://aclweb.org/anthology/D14-1162. [Accessed: May 07, 2023]
[14] F. Huber, 'King -Man +Woman = King ?', *Medium*, Jul. 15, 2019. [Online]. Available: https://blog.esciencecenter.nl/king-man-woman-king-9a7fd2935a85. [Accessed: May 12, 2023]
[15] S. P. and A. P. Shaji, 'A Survey on Semantic Similarity', in *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)*, Mumbai, India, Dec. 2019, pp. 1–8, doi: 10.1109/ICAC347590.2019.9036843 [Online]. Available: https://ieeexplore.ieee.org/document/9036843/. [Accessed: May 06, 2023]
[16] P. Resnik, 'Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language', 2011, doi: 10.48550/ARXIV.1105.5444. [Online]. Available: https://arxiv.org/abs/1105.5444. [Accessed: May 07, 2023]
[17] G. A. Miller, 'WordNet: a lexical database for English', *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995, doi: 10.1145/219717.219748.
[18] M. Sahlgren, 'The Distributional Hypothesis', *Ital. J. Linguist.*, 2008 [Online]. Available: https://www.semanticscholar.org/paper/The-Distributional-Hypothesis-Sahlgren/42aafb64d039235b84e1a6989302a318ba77c558. [Accessed: May 07, 2023]
[19] D. M. Blei, A. Y. Ng, and M. I. Jordan, 'Latent dirichlet allocation', *J. Mach. Learn. Res.*, vol. 3, no. null, pp. 993–1022, Mar. 2003.
[20] N. E. Evangelopoulos, 'Latent semantic analysis: Latent semantic analysis', *Wiley Interdiscip. Rev. Cogn. Sci.*, vol. 4, no. 6, pp. 683–692, Nov. 2013, doi: 10.1002/wcs.1254.
[21] 'Gensim: topic modelling for humans'. [Online]. Available: https://radimrehurek.com/gensim/. [Accessed: May 07, 2023]
[22] 'spaCy · Industrial-strength Natural Language Processing in Python'. [Online]. Available: https://spacy.io/. [Accessed: May 07, 2023]
[23] R. Python, 'Natural Language Processing With Python's NLTK Package – Real Python'. [Online]. Available: https://realpython.com/nltk-nlp-python/. [Accessed: May 07, 2023]
[24] 'IBM Watson'. [Online]. Available: https://cloud.ibm.com/developer/watson/dashboard. [Accessed: May 07, 2023]
[25] 'Cloud Natural Language', *Google Cloud*. [Online]. Available: https://cloud.google.com/natural-language. [Accessed: May 07, 2023]
[26] T. Tammet, 'Dataset of cooccurring words in Wikipedia'. [Online]. Available: http://turing.cs.ttu.ee/~Tanel.Tammet/wikicooccurrence.tar.gz. [Accessed: Dec. 05, 2023]
[27] R. Speer, J. Chin, and C. Havasi, 'ConceptNet 5.5: An Open Multilingual Graph of General Knowledge'. arXiv, Dec. 11, 2018 [Online]. Available: http://arxiv.org/abs/1612.03975. [Accessed: May 06, 2023]
[28] 'Numberbatch pre-computed word embeddings model'. commonsense, May 06, 2023 [Online]. Available: https://github.com/commonsense/conceptnet-numberbatch. [Accessed: May 07, 2023]

[29] ‘Facebook fastText library’. Meta Research, May 07, 2023 [Online]. Available: https://github.com/facebookresearch/fastText. [Accessed: May 07, 2023]
[30] M. Mihált, ‘word2vec-GoogleNews-vectors’. Apr. 18, 2023 [Online]. Available: https://github.com/mmihaltz/word2vec-GoogleNews-vectors. [Accessed: May 07, 2023]
[31] ‘BART pre-trained English language model’. [Online]. Available: https://huggingface.co/facebook/bart-large. [Accessed: May 07, 2023]
[32] ‘RoBERTa’. [Online]. Available: https://huggingface.co/docs/transformers/model_doc/roberta. [Accessed: May 07, 2023]
[33] C. Olah and S. Carter, ‘Attention and Augmented Recurrent Neural Networks’, *Distill*, vol. 1, no. 9, p. e1, Sep. 2016, doi: 10.23915/distill.00001.

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I Reimo Vellemaa

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis ”A Lexicon of Semantic Similarity and Difference: An Analysis of the 20 000 Most Common English Words”, supervised by Martin Verrev and Tanel Tammet
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

1 The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.