TALLINNA TEHNIKAÜLIKOOL
INSENERITEADUSKOND
Virumaa kolledž

Pythoni baasil sotsiaalvõrkude tekstianalüüsi meetodite rakendamine TalTech Virumaa kolledži Facebooki näitel

Application of Python-based social networks text analysis methods on the example of Facebook of TalTech Virumaa College

RAKENDUSINFOTEHNOLOOGIA ÕPPEKAVA LÕPUTÖÖ

Üliõpilane: Dmitri Rõbovalov
Üliõpilaskood: 121357
Juhendaja: Olga Dunajeva, lektor

Kohtla-Järve, 2021

TALLINNA TEHNIKAÜLIKOOL
INSENERITEADUSKOND
Virumaa kolledž

Применение основанных на Python методов анализа текстовых данных социальных сетей на примере Facebook-страницы Вирумааского колледжа ТТУ

RAKENDUSINFOTEHNOLOOGIA ÕPPEKAVA LÕPUTÖÖ

Üliõpilane: Dmitri Rõbovalov
Üliõpilaskood: 121357
Juhendaja: Olga Dunajeva, lektor

Kohtla-Järve, 2021

AUTORIDEKLARATSIOON

Olen koostanud lõputöö iseseisvalt.

Lõputöö alusel ei ole varem kutse- või teaduskraadi või inseneridiplomit taotletud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

"15" mai 2021.

Autor: Dmitri Rõbovalov
/ allkiri /

Töö vastab rakenduskõrgharidusõppe lõputööle/magistritööle esitatud nõuetele "15" mai 2021.

Juhendaja: Olga Dunajeva
/ allkiri /

Kaitsmisele lubatud
"...." ..... 20.....

Kaitsmiskomisjoni esimees .....
/ nimi ja allkiri /

LIHTLITSENTS LÕPUTÖÖ ÜLDSUSELE KÄTTESAADAVAKS TEGEMISEKS JA REPRODUTSEERIMISEKS

Mina Dmitri Rõbovalov (sünnikuupäev: 24.11.1990)

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose

"Pythoni baasil sotsiaalvõrkude tekstianalüüsi meetodite rakendamine TalTech Virumaa kolledži Facebooki näitel", mille juhendaja on Olga Dunajeva,

1.1. reprodutseerimiseks säilitamise ja elektroonilise avaldamise eesmärgil, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. Olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. Kinnitan, et lihtlitsentsi andmisega ei rikuta kolmandate isikute intellektuaalomandi ega isikuandmete kaitse seadusest ja teistest õigusaktidest tulenevaid õigusi.

TalTech Inseneriteaduskond Virumaa kolledž

LÕPUTÖÖ ÜLESANNE

Üliõpilane: Dmitri Rõbovalov, 121357RDIR

Õppekava, peaeriala: RDIR02/12 - Rakendusinfotehnoloogia

Juhendaja: Lektor, Olga Dunajeva, olga.dunaeva@taltech.ee

Konsultant: puudub

Lõputöö teema:

Pythoni baasil sotsiaalvõrkude tekstianalüüsi meetodite rakendamine TalTech Virumaa kolledži Facebooki näitel

Application of Python-based social networks text analysis methods on the example of Facebook of TalTech Virumaa College

Lõputöö põhieesmärgid:

1. Sotsiaalvõrkude andmete hankimise võimaluste uurimine ja sobiva meetodi valimine Virumaa kolledži Facebooki andmete kogumiseks.
2. Sotsiaalvõrkude tekstiandmete analüüsimeetodite uurimine ja Virumaa kolledži Facebooki andmete analüüsimiseks sobivate Pythoni-põhiste meetodite valimine.
3. Valitud meetodite rakendamine Virumaa kolledži Facebooki andmete kogumiseks ja analüüsimiseks, analüüsitulemustest järelduste tegemine.

Lõputöö etapid ja ajakava:

Nr	Ülesande kirjeldus	Tähtaeg
1.	Sotsiaalvõrkude andmete hankimise võimaluste uurimine, sobiva meetodi valimine, Virumaa kolledži Facebooki andmete kogumine.	10.03.2021
2.	Lõputöö struktuuri loomine. Sissejuhatuse kirjutamine	15.03.2021
3.	Sotsiaalvõrkude tekstiandmete analüüsimeetodite uurimine, Virumaa kolledži Facebooki andmete analüüsimiseks sobivate Pythoni-põhiste meetodite valimine.	28.03.2021
4.	Virumaa kolledži Facebooki andmete analüüs. Lõputöö põhiosa kirjutamine.	18.04.2021

5.	Järelduste tegemine, kokkuvõtte kirjutamine, lõputöö lõplik vormistamine.	16.05.2021

Töö keel: vene keel **Lõputöö esitamise tähtaeg:** "28" detsember 2021 a

Üliõpilane: ..... "....."..... 20.....a
/allkiri/

Juhendaja: ..... "....."..... 20.....a
/allkiri/

Konsultant: ..... "....."..... 20.....a
/allkiri/

Programmijuht: ..... "....."..... 20.....a
/allkiri/

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ

Тема была обдуманная и согласованна с лектором Вирумааского колледжа Таллиннского Технического Университета Ольгой Дунаевой.

Все материалы и данные по изучению выбранной темы были найдены автором самостоятельно.

В данной работе проводится анализ текста с определением тематики постов официальной страницы Facebook Вирумааского колледжа Таллиннского Технического Университета https://www.facebook.com/TalTechVK.

Ключевые слова: анализ текстовых данных, социальная сеть, Python, Facebook, дипломная работа.

СПИСОК СОКРАЩЕНИЙ И СИМВОЛОВ

ПО – Программное обеспечение.

API – Application programming interface

FS – Facebook Scraper

FB – Facebook

JSON – Текстовый формат обмена данными

NLTK – Natural Language Toolkit

Open-source - Открытое программное обеспечение

Sentiment analysis – Анализ тональности текста

TW – Twitter

URL - Uniform Resource Locator

VK – Vkontakte

ВВЕДЕНИЕ

Тема данной выпускной работы – применение методов анализа текстовых данных социальных сетей на базе языка программирования Python на примере Facebook-страницы Вирумааского колледжа Таллиннского Технического Университета.

Социальные сети в последнее время развиваются с большой скоростью. С каждым днем становится все больше и больше пользователей, которые хотели бы стать участником социальной сети. Социальные сети уже давно стали обыденностью нашей жизни. Мы контактируем с ними каждый день, и для большинства людей социальные сети стали платформой для коммуникаций.

Современное программное обеспечение позволяет извлекать содержимое со страниц социальной сети и проводить над ним анализ. В настоящий момент существуют алгоритмы, позволяющие определять принадлежности текста к тем или иным тематикам по различным методикам. Благодаря этому можно получить информацию о востребованных тематиках среди различной аудитории.

Готовых полноценных (выгрузка данных, перевод, предварительная обработка и анализ текста) решений для определения тематики текстов на эстонском и русском языках на настоящий момент нет. Имеются ряд модулей Python и проектов с открытым исходным кодом на платформе GitHub, однако они имеют ряд недостатков применительно к настоящей выпускной работе: отсутствует поддержка эстонского языка, используется Facebook API, доступ к которому существенно ограничен.

Целью настоящей выпускной работы является изучение основанных на Python методов сбора и анализа текстовых данных социальных сетей и их применение для анализа страницы Facebook Вирумааского колледжа ТТУ.

Для достижения поставленной цели автору необходимо было решить следующие задачи:

- выбор технологии извлечения содержимого постов страницы социальной сети Facebook;
- сбор текстовых данных страницы Facebook Вирумааского колледжа ТТУ;
- выбор технологии машинного перевода текста и его применение;
- проведение предварительной обработки текстов для дальнейшего анализа;
- выбор алгоритмов анализа текстовых данных;
- анализ текстовых данных страницы Facebook Вирумааского колледжа.

Анализ социальных сетей в настоящее время не является чем-то новым в сфере информационных технологий. Разработчики социальных сетей Twitter, Facebook,

Vkontakte для удобства работы конечных пользователи разработали соответствующие API для доступа к этим социальным сетям. Например, для компании Twitter это Twitter API, для Facebook это Facebook API, для Vkontakte это Vk API. Стоит отметить, что с 2018 года Facebook существенно ограничил использование своего API: для получения доступа к API нужно пройти многоэтапную проверку в Facebook, чтобы обосновать цели использования. Были произведены неоднократные попытки на получение доступа к API, однако во всех случаях был получен отказ без объяснения причин.

В связи с этим было принято решения провести поиск альтернативных технологий извлечения информации из Facebook. Среди них была выделена технология Веб-скрейпинга. Проект с открытым исходным кодом, которой будет изучен и применен в настоящей выпускной работе, называется «Facebook Scraper», разработан специально для извлечения информации о постах со страниц социальной сети Facebook.

Машинный перевод извлеченного текста проводился с использованием сервиса Google Translate. Предварительная обработка осуществлялась с использованием библиотеки NLTK. Для анализа тематик постов из социальной сети был разработан алгоритм определения тематики текста с использованием словаря и правил. Для реализации алгоритмов использовался язык программирования Python в интегрированной среде разработки PyCharm.

Структура данной выпускной работы выглядит следующим образом.

В первой главе представлены теоретические аспекты работы с текстовыми данными. Описано то, каким способом будет извлекаться информация, каким способом будет выполняться перевод информации, как сделать предварительную обработку текста, а также описаны методы по определению тематики.

Во второй главе описана практическая часть по извлечению информации, переводу текста, и подготовки данных для запуска алгоритмов определения тематики.

В третьей главе представлена практическая часть по применению алгоритмов, которые определят тематику текста, а также представлены результаты определения тематик и их анализ.

1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ОБРАБОТКИ И АНАЛИЗА ТЕКСТА

1.1 Источники данных для обработки

Источники данных существуют в разной форме. Например, это могут быть социальные сети, различные базы данных (уже с выгруженными постами/статьями/текстами), разные статьи на страницах интернет-ресурсов, новостные сводки. Источником данных для анализа текста в настоящей выпускной работе будет одна из популярных социальных сетей – Facebook, а именно официальная страница Вирумааского колледжа Таллиннского Технического Университета.

Как известно, любая веб-страница создаётся с использованием языков разметки (XHTML и HTML), которые содержат интересующую информацию в программном коде. Чтобы это увидеть, достаточно открыть исходный код интересующей страницы и найти в нём необходимую информацию – текст поста, лайки/дизлайки, отметки о репостах, даты создания постов и т. д. Важно отметить, что исходный код публичных страниц социальной сети Facebook достаточно сложен для визуального анализа и поиска подобной информации в связи с большим объемом исходного кода, написанного на языках HTML, Javascript, AJAX и т. д. Таким образом, для извлечения информации со страниц Facebook необходимо применение средств автоматизации процесса поиска и получения данных о постах.

Для извлечения информации со станицы колледжа, как говорилось ранее, будет использоваться Веб-скрейпинг. Веб-скрейпинг – это технология, позволяющая получать веб-данные путем извлечения информации из HTML-разметки страницы, имитируя работу веб-браузера. Операцию по извлечению информации о постах со страницы социальной сети Facebook будет выполнять код, который будет отправлять запросы (HTTP GET) на целевой сайт и получать HTTP ответ с запрошенными данными (текст поста, дата, количество лайков/репостов и т. д.).

1.2 Машинный перевод

Существуют различные способы перевода текстовой информации. В целом можно выделить два способа: ручной и машинный переводы. По качеству перевода, безусловно, предпочтительнее ручной перевод текста, но, когда информации становится достаточно много, то появляется необходимость в автоматизации этого процесса. Одной из самых удобных и быстрых технологий перевода является использование API какой-либо платформы машинного перевода. Наиболее

известными платформами являются Google API Translate и Yandex API Translate. Эти платформы пользуются большой популярностью и работают с десятками языков, автоматически определяя тематики текстов для более качественного перевода.

Однако стоит отметить, что у каждой из этих платформ есть свои ограничения. Так, Yandex с 2020 года прекратил бесплатное предоставление API для сервиса машинного перевода. Google, в свою очередь, распространяет функции перевода текстов бесплатно, но с существенными ограничениями на размер текста для перевода и количество переводов в сутки. Это может стать проблемой при переводе большого количества информации.

В связи с существующими ограничениями API от сервисов Google и Yandex было принято решение о поиске альтернативной технологии машинного перевода. В качестве такой альтернативы был выделен способ перевода, схожий с технологией Веб-скрейпинга. Принцип работы такого алгоритма заключается в следующем:

- Формируется HTTP-запрос на перевод какого-либо отрывка текста. Запрос формируется максимально схожим с тем, который генерирует веб-браузер;
- Запрос отправляется на портал машинного перевода: Yandex.Translate или Google.Translate;
- Ожидается ответ на отправленный запрос. Ответ будет содержать фактически веб-страницу с переведенным текстом;
- Осуществляется извлечение с помощью регулярных выражений переведенного текста из HTML-разметки полученной веб-страницы.

1.3 Предварительная обработка текста

Анализ текста на принадлежность той или иной тематике будет реализован с применением известного подхода «Bag-of-words» - модели, опирающейся на количество вхождений слов в данный текст [1]. Для применения этого подхода потребуется предварительная обработка текста, которая называется нормализацией и состоит из следующих шагов:

- нормализация – приведение слов к нижнему регистру, удаление знаков пунктуации, специальных символов, чисел;
- токенизация – разбиение текста на слова;
- удаление стоп-слов – слов, которые не несут смысловой нагрузки: артиклей, союзов, предлогов, междометий;
- лемматизация - приведение слова к его смысловой канонической форме (инфинитив несовершенного вида для глаголов, именительный падеж единственного числа мужского рода — для существительных и

прилагательных). [2]

Схема предварительной обработки текста с примером предварительной обработки предложения представлена на Рисунке 1.1.

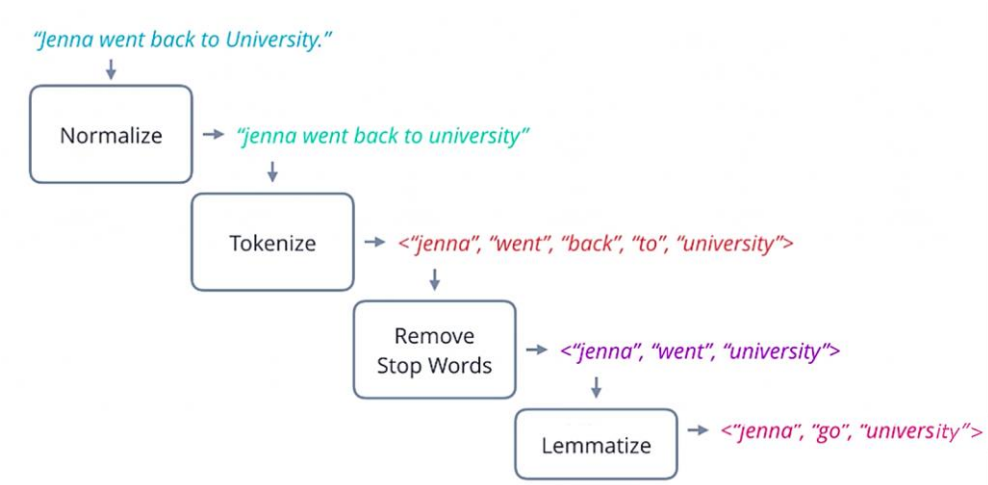

Рисунок 1.1 Схема предварительной обработки текста [2]

1.4 Анализ текстовых данных: определение тематики

На настоящий момент существуют различные способы автоматического определения принадлежности текста к той или иной тематике. Выделим наиболее распространенные методики:

- анализ, основанный на правилах;
- анализ по словарю;
- анализ с применением машинного обучения. [3]

Анализ тематики текста, основанный на правилах, является наиболее точным, но, в то же время наиболее трудоемким в плане реализации. Анализ по словарю обычно показывает хорошие результаты, но имеет особенность: методика не является универсальной, необходимо готовить словарь для каждой рассматриваемой предметной области. Анализ текстов с применением машинного обучения хорошо себя показывает в области определения тональности, где фактически возможных тематик две – хорошая и плохая. В случае, когда возможных тематик более двух, этот подход становится весьма трудоемким для реализации и не всегда показывает хорошие результаты.

В ходе выборочного визуального анализа содержимого постов страницы колледжа были выявлены следующие особенности:

- среди постов можно выделить следующие тематики: *Курсы/тренинги/мероприятия*, *Прием на работу/учёбу*, *Текущая учеба*,

Наука и техника, *Происшествия*;

- посты разных тематик зачастую очень близки по содержимому: если в этих тематиках выделить ключевые слова, то они будут похожи по смыслу либо будут одинаковыми.

В связи с выявленными особенностями применение машинного обучения для данной предметной области не представляется эффективным, поскольку из-за высокой схожести содержимого постов разных тематик алгоритмы с машинным обучением будут давать большое количество ошибочных результатов. Кроме того, существенным ограничением является отсутствие достаточного количества данных для обучения классификатора, поскольку предполагалось бы в качестве обучающего набора рассматривать посты за 2021 год. Посты за 2019 и 2020 годы рассматривать в качестве обучающей выборки было бы некорректно, так как эти посты необходимо анализировать на обученном классификаторе.

С целью увеличения эффективности определения тематик постов страницы колледжа, автором настоящей работы было принято решение о построении алгоритма с применением двух методик: по словарю и на основе правил. Планируется разработать алгоритм, который будет использовать сильные стороны и преимущества каждой из них.

1.4.1 Алгоритм LDA (Latent Dirichlet Allocation)

Латентное размещение Дирихле (Latent Dirichlet Allocation, LDA) - метод тематического моделирования, который был предложен Дэвидом Блеем (David Blei), Эндрю Ыном (Andrew Ng) и Майклом Джорданом (Michael Jordan) в 2003 г. LDA принадлежит семейству, порождающий вероятностных моделей, в которых темы представлены вероятностями появления каждого слова из заданного набора. Документы в свою очередь могут быть представлены как сочетания тем. Особенность моделей LDA состоит в том, что темы не обязательно должны быть различными и слова могут встречаться в нескольких темах; это придает некоторую нечеткость определяемым темам, что может пригодиться для совладения с гибкостью языка. [4]

1.5 Инструменты и средства

Автор данной работы будет использовать следующие средства разработки:

- Язык программирования Python версии 3.8 [5],
- Интегрированная среда разработки Pycharm [6].

Так же будут использоваться следующие библиотеки и средства Python: NTLK, Gensim, Matplotlib, Pymorphy2, Urllib, Emmet, Wordcloud, re.

NLTK – Natural Language Toolkit, пакет библиотек для обработки текста. [7]

Gensim – это библиотека с открытым исходным кодом для тематического моделирования и обработки естественного языка с использованием современного статистического машинного обучения. [8]

Matplotlib – библиотека для визуализации данных (диаграммы, графики). [9]

Pymorphy2 – морфологический анализатор для русского языка. [10]

Urllib – модуль для открытия URL-адресов. [11]

Emmet – набор инструментов для работы с HTML-разметкой. [12]

Wordcloud – библиотека для создания облака слов. [13]

Re – модуль для операций с регулярными выражениями. [14]

Данные библиотеки были использованы как одни из самых популярных для языка Python (вывод был сделан на основе анализа открытых проектов по обработке текстов).

Urllib, Emmet – это стандартные модули языка Python, которые используют программисты как стандарт де-факто при работе с HTML-разметкой и URL-адресами.

2 ПРЕДВАРИТЕЛЬНАЯ ОБРАБОТКА ТЕКСТА

2.1 Извлечение данных для анализа

Как говорилось ранее, для извлечения данных с официальной страницы колледжа в социальной сети Facebook был использован модуль FacebookScrapper, разработанный для языка программирования Python. Для установки модуля необходимо в окружении проекта выполнить консольную команду (см Рисунок 2.1)

```
pip install facebook-scraper
```

Рисунок 2.1 Установка модуля FacebookScrapper

После успешной установки необходимо подключить установленный модуль к проекту (см Рисунок 2.2).

```
from facebook_scraper import get_posts.
```

Рисунок 2.2 Подключение модуля к проекту

После выполнения этих действий приложение готово к использованию и извлечению информации о постах. FacebookScrapper работает подобно браузеру и в фоновом режиме осуществляет вход на указанную веб-страницу социальной сети Facebook. Основной метод get_posts модуля FacebookScrapper принимает два параметра: идентификатор публичной страницы и количество страниц, которое будет пролистано для получения постов. Метод get_posts, используя HTML-разметку страницы, осуществляет извлечение содержимого постов целевой группы (см Рисунок 2.3).

```
posts = get_posts(page_name, pages=220)
```

Рисунок 2.3 Метод get_posts

Важно отметить, что число страниц, которые необходимо пролистать в целевой группе социальной сети Facebook достигает 200–300 штук. Подобная процедура может занять порядка получаса, и выполнять ее каждый раз при запуске приложения крайне неэффективно и кроме того, частое выполнение подобных действий может стать причиной блокировки IP адреса, с которого ведется выгрузка содержимого постов. В связи с этим было принято решение сохранить выгруженные посты в файл, чтобы ускорить дальнейшую обработку текстовых данных.

Для сохранения данных использовался формат JSON, поскольку является весьма удобным для работы на языке программирования Python. При использовании JSON формата, Python автоматически создает необходимы структуры данных (списки,

словари) при загрузке данных из файла, существенно ускоряя процесс написания кода. На рисунке 2.4 представлена функция, которая осуществляет выгрузку постов за 2019–2021 годы и сохраняет их в файл формата JSON.

```
def exract_posts(page_name, save_to):
    out = open(save_to, "w", errors="ignore")
    posts = get_posts(page_name, pages=220)
    #Проход по постам и извлечение только необходимых полей у постов
    for p in posts:
        # Формирование объекта JSON для последующего сохр в файл с именем save_to
        data = {}
        data["post_id"] = p["post_id"]
        data["text"] = p["text"]
        data["likes"] = p["likes"]
        data["comments"] = p["comments"]
        data["shares"] = p["shares"]
        data["time"] = (p["time"]).strftime("%Y-%m-%d")
        #Запись в файл
        json.dump(data, out)
        # Переход на новую строку
        out.write("\n")
    out.close()
```

Рисунок 2.4 Функция выгрузки постов

2.2 Реализация машинного перевода текста

Для дальнейшей подготовки данных для обработки требуется средство автоматизированного перевода. Как упоминалось ранее, использовалась методика, подобная работе FacebookScrapper, то есть имитировалась работа браузера при использовании сервиса Google Translate.

Алгоритм перевода текстов с использованием сервиса Google Translate был заимствован из open-source проекта [15]. Указанный проект имеет функцию translate, которая принимает на вход три параметра: текст для перевода, исходный язык, целевой язык. Для использования этой функции требуется подключение следующих модулей (см Рисунок 2.5).

```
import html
import urllib.request
import urllib.parse
```

Рисунок 2.5 Подключаемые модули для реализации машинного перевода

Алгоритм перевода заключается в следующих шагах:

- Подготовка ссылки на ресурс Google-translate (см Рисунок 2.6).

```
base_link = "http://translate.google.com/m?tl=%s&sl=%s&q=%s"
```

Рисунок 2.6 Ссылка на ресурс Google-translate

- Затем вместо параметров %s в ссылке подставляются значения исходного языка, языка назначения и текста для перевода (см Рисунок 2.7).

```
link = base_link % (to_language, from_language, to_translate)
```

Рисунок 2.7 Ссылка с параметрами

- Далее формируется объект Request, содержащий ссылку и заголовки браузера Mozilla Firefox, и осуществляется его отправка с последующим ожиданием ответа от сервера Google (см Рисунок 2.8).

```
request = urllib2.Request(link, headers=agent)
raw_data = urllib2.urlopen(request).read()
```

Рисунок 2.8 Создание запроса и его отправка

- Финальным этапом алгоритма является процесс извлечения переведенного текста из полученной HTML-страницы с сервиса Google-translate (см Рисунок 2.9).

```
data = raw_data.decode("utf-8")
```

Рисунок 2.9 Процесс извлечения переведенного текста

2.3 Реализация предварительной обработки текста

Предварительная обработка текста состоит из трёх этапов:

- нормализация и токенизация - разбиение предложений на отдельные слова с удалением знаков препинания, смайлов, эмодзи и прочих символов;
- удаление стоп-слов - слов, часть речи которых не представляет значимости для анализа тематики текстов;
- лемматизация оставшихся слов.

2.3.1 Удаление несловесных конструкций

Для удаления ненужных символов таких как смайлики, эмодзи и прочих мини-картинок использовалось регулярное выражение, представленное на рисунке 2.11.

Для создание регулярного выражения необходимо воспользоваться модулем re, который нужно импортировать в проект (см Рисунок 2.10).

```
import re
```

Рисунок 2.10 Импортирование модуля re

```
emoji_removing_pattern = re.compile("["
                        u"\U0001F600-\U0001F64F"  # emoticons
                        u"\U0001F300-\U0001F5FF"  # symbols & pictographs
                        u"\U0001F680-\U0001F6FF"  # transport & map symbols
                        u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
                        u"\U0001F1F2-\U0001F1F4"  # Macau flag
                        u"\U0001F1E6-\U0001F1FF"  # flags
                        "]+", flags=re.UNICODE)
```

Рисунок 2.11 Удаление ненужных символов

Указанные в комментариях группы символов были заменены на пустую строку: “”. Код каждого символа в предложении сравнивался с кодом символа, подлежащего удалению, и если обнаруживалось совпадение, то символ удалялся (см Рисунок 2.12)

```
post_without_emojies = emoji_removing_pattern.sub(r'', post["text"])
```

Рисунок 2.12 Процесс удаления ненужных символов

2.3.2 Токенизация и удаление стоп-слов

Для тонизации и удаления ненужных слов для последующего анализа частей речи использовался пакет библиотек и программ для символьной и статистической обработки естественного языка NLTK. В пакете NLTK был использован метод word_tokenize, который осуществляет разбиение предложения на отдельные слова, удаляя все знаки препинания (см Рисунок 2.13).

```
words = nltk.word_tokenize(post_without_emojies)
```

Рисунок 2.13 Метод разбиения предложения на отдельные слова

Далее каждому слову была сопоставлена его часть речи для того, чтобы удалить из предложения слова, часть речи которых несущественна для дальнейшего анализа (см Рисунок 2.14).

```
part_of_speech_toremove = {'ADV', 'A-PRO','A-PRO=m',
                           'A-PRO=n','ADV-PRO','A-PRO=pl',
                           'CONJ', 'INTJ','NONLEX',
                           'PART', 'PR', 'PRAEDIC-PRO',
                          'PRAEDIC','PARENTH', 'S-PRO'
                          }
```

Рисунок 2.14 Список частей речи для удаления

В указанном выше фрагменте ADV – наречие; CONJ – союз; PRO – местоимение; PR – предлоги; NONLEX – различные символы; PARENTH – вводные слова, PART – частицы; A-PRO, ADV-PRO, S-PRO, PRAEDIC – несущественные части, такие как местоимение-прилагательное, местоименное наречие, местоимение-существительное и т.п.

В приведенной ниже операции осуществляется фильтрация тех слов, часть речи которых присутствует в списке для удаления (см Рисунок 2.15).

```
post["text"] = [word for word, pos in nltk.pos_tag(words, lang='rus') if pos not in part_of_speech_toremove]
```

Рисунок 2.15 Операция для фильтрации текста по частям речи

Другими словами, остаются только слова, часть речи которых является существенной для проведения последующего анализа текста на принадлежность к той или иной тематике.

Для удаления стоп слов использовались встроенные и самостоятельно разработанные словари. Которые приведены на рисунке 2.16

```
151         if self.dest_lang == 'ru':
152             if self.for_LDA:
153                 with open(os.path.join(current_dir, "stop_words", "rus.txt"), 'r',encoding="utf-8") as fin:
154                     stop_words = fin.read().splitlines()
155             else:
156                 stop_words = stopwords.words('russian')
157         elif self.dest_lang == 'en':
158             with open(os.path.join(current_dir,"stop_words","eng.txt"),'r',encoding="utf-8") as fin:
159                 stop_words = fin.read().splitlines()
160             #stop_words = stopwords.words('english')
161             stop_words += list('.,;:#$%!?%@&*"`\'-·+=)([]}{|')
162             stop_words += [ '\'s','``','\'\'','vkg','--','al','la','\'m','tee']
```

Рисунок 2.16 подготовка словарей стоп слов.

2.3.3 Лемматизация слов

Заключительным этапом предварительной обработки текста является приведение слова к его нормальной форме или лемматизация.

Этот этап предлагается оформить в виде метода normalize_words() класса DataPreparer. Ввиду большого объема, исходный код метода приведен в Приложении 1. Этот метод имеет разную логику работы в зависимости от языка, на котором обрабатываются текстовые данные. Первый шаг – загрузка списков стоп-слов и их дополнение в зависимости от целевого языка. Второй шаг – проход по содержимому постов, преобразование каждого слова к начальной форме (лемматизация) и проверка полученного слова на вхождение в список стоп-слов. Отметим, для английского и русского языков используются разные модули лемматизации.

Для этого использовался пакет pymorphy2 который содержит в себе морфологический анализатор MorphAnalyzer и NLTK, который содержит в себе анализатор WordNetLemmatizer. Процесс приведения слова к его нормальной форме сравнительно прост: для этого создается объект MorphAnalyzer, вызывается его метод parse() с передачей интересующего слова в качестве параметра, после чего можно получить доступ к его нормальной форме. На рисунке 2.17 приведен исходный код процесса нормализации слов в выгруженных постах на русском языке.

```
morph = pymorphy2.MorphAnalyzer()
for post in self.posts:
    normalized_post = []
    for word in post["text"]:
        #получить информацию о слове word (все возможные разборы слова)
        #[0] - взятие наиболее вероятного разбора
        p = morph.parse(word)[0]
        #взятие нормальной формы
        word_normal = p.normal_form
        #Добавление нормальной формы слова в пост
        normalized_post.append(word_normal)
    post["text"] = normalized_post
```

Рисунок 2.17 Процесс нормализации слов

2.3.4 Подготовленные данные

Анализ текста был произведен на 686 постах. Длина поста вариировалась от 4 до 455 слов. В среднем длина поста равнялась 55 словам. Ниже на Рисунках 2.18 и 2.19 приведены облака ключевых слов для постов официальной Facebook страницы колледжа на английском языке за 2019 и 2020 год соответственно. Для построения облака тегов (ключевых слов) использовалась библиотека wordcloud языка Python. Ключевые слова выбирались на основе частоты их встречаемости (наиболее частые слова) в постах за 2019 и 2020 год.

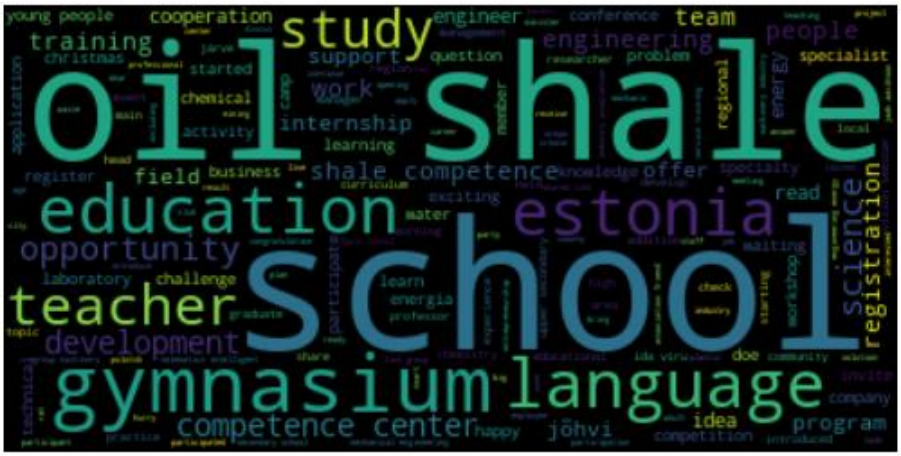

Рисунок 2.18. Ключевые слова для постов за 2019 год (EN)

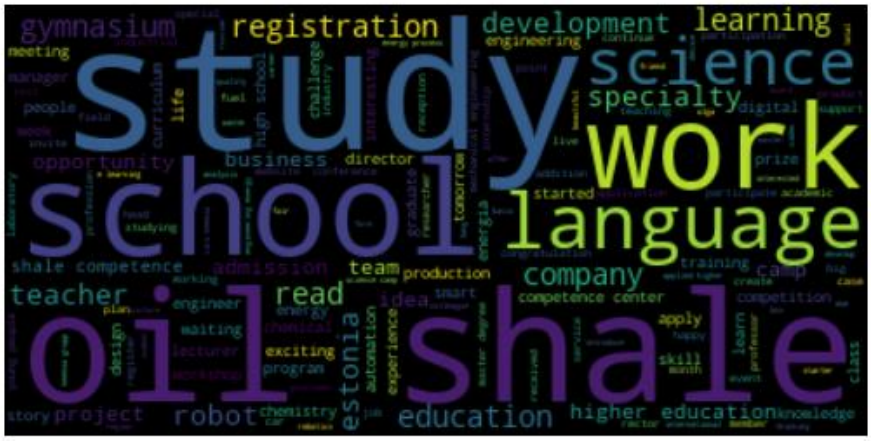

Рисунок 2.19. Ключевые слова для постов за 2020 год (EN)

Общее количество слов в постах за 2019 и 2020 год составило 37902 слова. После всех этапов предварительной обработки это количество уменьшилось до 20462 слов, то есть практически половина слов были помечены как излишние для проведения анализа.

Ниже представлены две картинки пост до обработки (Рисунок 2.20) и пост после обработки (Рисунок 2.21)

```
Студенты факультета информационных технологий и машиностроения Вирумааского колледжа
практикуют совместную работу, решая реальные проблемы. Сегодня они ищут решения двух тем,
предлагаемых Центральной больницей Ида-Виру: разработка системы контроля температуры и
моделирование, калибровка и сборка коллиматора.
👇 Подробнее:
https://taltech.ee/uudised/taltech-virumaa-kolledzi-tudengid-otsivad-lahendusi-ida-viru-kes
khaigla-susteemide

TALTECH.EE
Студенты колледжа TalTech Virumaa ищут решения для модернизации систем центральной
больницы Ида-Виру | TalTech
```

Рисунок 2.20 Пост до обработки

```
['студент', 'факультет', 'информационный', 'технология', 'машиностроение', 'вирумааский', 'колледж',
'практиковать', 'совместный', 'работа', 'решать', 'реальный', 'проблема', 'искать', 'решение',
'предлагать', 'центральный', 'больница', 'ида-вира', 'разработка', 'система', 'контроль', 'температура',
'моделирование', 'калибровка', 'сборка', 'коллиматор', 'подробный', 'студент', 'колледж', 'искать',
'решение', 'модернизация', 'система', 'центральный', 'больница', 'ида-вира']
```

Рисунок 2.21 Пост после обработки

3 РЕАЛИЗАЦИЯ АЛГОРИТМА ОПРЕДЕЛЕНИЯ ТЕМАТИКИ ТЕКСТА

3.1 Алгоритм LDA

Для применения алгоритма LDA была выбрана библиотека Gensim языка программирования Python, поскольку модели в Gensim имеют больше настраиваемых параметров, чем в модуле scikit-learn. Кроме того, Gensim изначально разрабатывалась как библиотека для тематического моделирования.

Для запуска алгоритма необходимо подготовить следующие данные:

- набор предварительно обработанных текстов для анализа;
- требуемое количество тем, которые алгоритм будет выявлять;
- ряд вспомогательных числовых параметров для более тонкой настройки модели.

Ниже приведены шаги, осуществляющие запуск алгоритма LDA на предварительно обработанных текстах.

3.1.1 Подготовка текстовых данных

На Рисунке 3.1 изображен фрагмент функции, использующейся для запуска алгоритма LDA. В начале ее работы формируется список из текстов постов, выгруженных со страницы колледжа. Далее на основе этого списка формируется словарь, содержащий пары вида: (word, id).

```
def run(normal_posts, translated_posts, lda_predicted_fname, num_topics):
    text_data = []
    for post in normal_posts:
        text_data.append(post["text"])

    dictionary = corpora.Dictionary(text_data)
```

Рисунок 3.1 Процесс формирования списка и словаря

Следующим шагом формируется объект-корпус, который уже непосредственно подается на вход функции LDA (Рисунок 3.2).

```
corpus = [dictionary.doc2bow(text) for text in text_data]
```

Рисунок 3.2 Формирование объекта-корпуса для подачи на вход функции LDA

3.1.2 Определение оптимального количества тем

Количество тем – достаточно важный параметр алгоритма LDA. При слишком малом значении алгоритм может упустить некоторые тематики, при слишком большом – тематики будут повторяться.

Существует подход, позволяющий определить оптимальное число тематик. Согласно нему, необходимо найти максимальное значение коэффициента согласованности тематик [16]. Ниже на Рисунке 3.3 представлена функция, определяющая коэффициент согласованности для заданного числа тематик.

```
@staticmethod
def show_optimal_topics_count(normal_posts,start,limit,step):
    text_data = []
    for post in normal_posts:
        text_data.append(post["text"])
    dictionary = corpora.Dictionary(text_data)
    corpus = [dictionary.doc2bow(text) for text in text_data]

    model_list, coherence_values = \
        LDAClassifier.compute_coherence_values(dictionary,
                                               corpus,
                                               text_data,
                                               limit=limit,
                                               start=start, step=step)
    limit = 20
    start = 2
    step = 2
    x = range(start, limit, step)
    plt.plot(x, coherence_values)
    plt.xlabel("Num Topics")
    plt.ylabel("Coherence score")
    plt.legend(("coherence_values"), loc='best')
    plt.show()
```

Рисунок 3.3 Формирование объекта-корпуса для подачи на вход функции LDA

На Рисунке 3.4 приведены результаты работы функции при запуске на выгруженных постах на русском языке.

По графику видно, что коэффициент согласованности достигает максимума на N=10 и 14 тем. Далее коэффициент начинает падать.

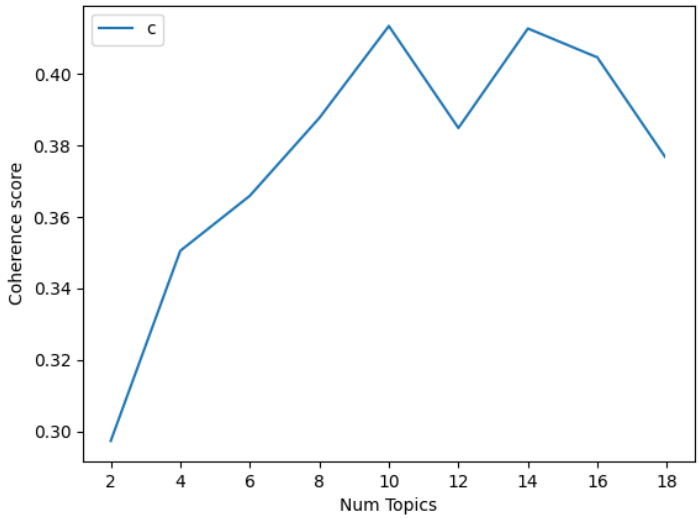

Рисунок 3.4 График коэффициента согласованности для количества тем N=2-18

Таким образом оптимальное количество тем над которым стоит проводить анализ, является 10 или 14. Для этого алгоритм LDA включает в себя сравнительную визуализацию, по которой можно наглядно видеть, какие темы пересекаются близко по смыслу. На Рисунке 3.5 Представлена сравнительная визуализация по 10 темам на русском языке.

Рисунок 3.5 Сравнительная визуализация по 10 темам на русском языке

На Рисунке 3.5 видно, что тема 3, близко связана с темами 7 и 2. Это можно наблюдать, по ключевым словам, которые опередил алгоритм LDA к данным темам. К теме номер 3 – Комитет по горючему сланцу относятся такие слова как: Технология, центр, сланец. В теме номер 7 и 2 присутствуют основные ключевые

слова такие как: исследования, технологический, сланец. Поэтому темы 7,3,2 взяты за одну общую тему – Комитет по горючему сланцу.

На Рисунке 3.6 приведены результаты работы функции при запуске на выгруженных постах на английском языке для количества тем 2–20.

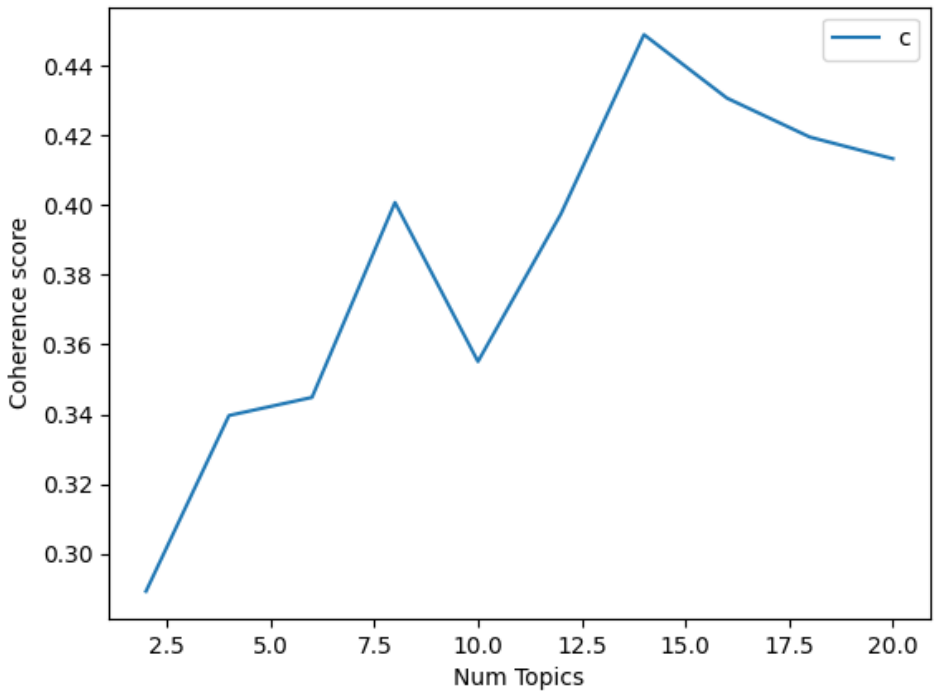

Рисунок 3.6 График коэффициента согласованности для количества тем N=2-20

На графике видно, что значение коэффициента достигает максимума на количестве тем N=14. Таким образом оптимальное количество тем над которым стоит проводить анализ на английском языке, является 14. На Рисунке 3.7 представлена визуализация по 14 темам на английском языке.

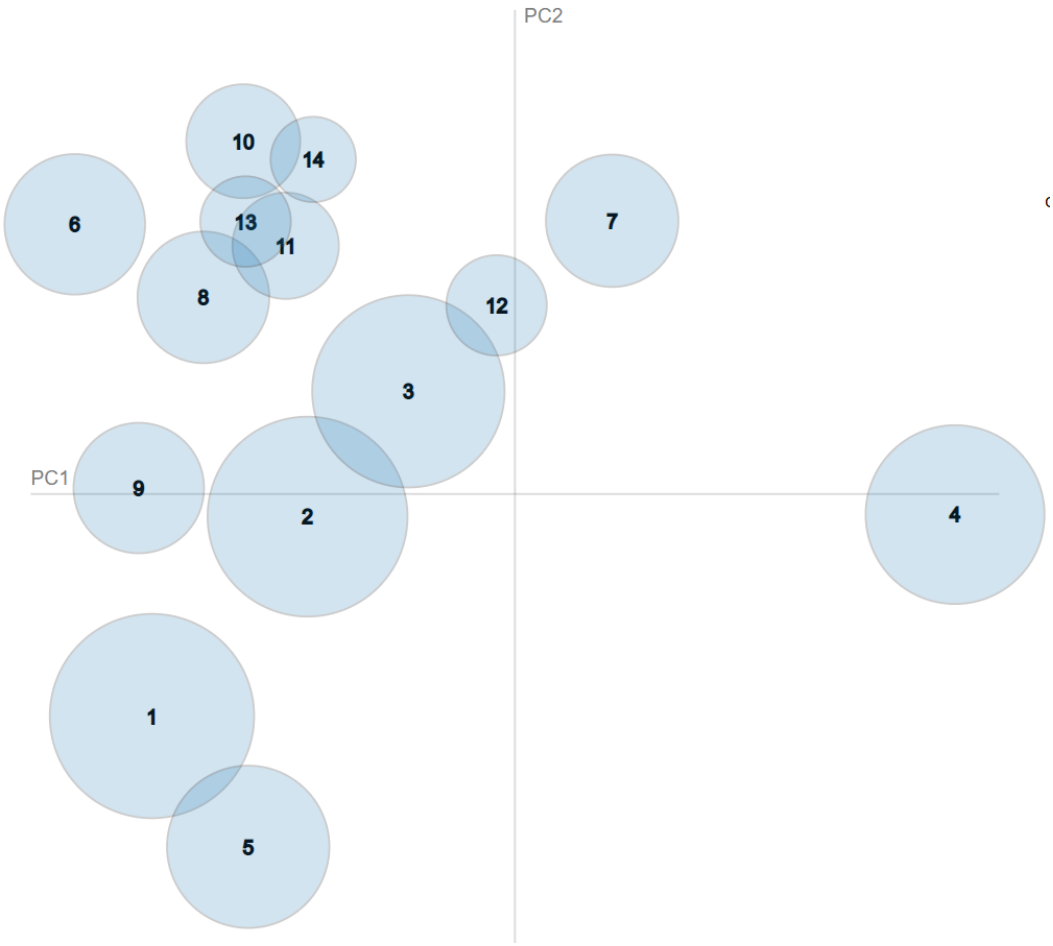

Рисунок 3.7 Сравнительная визуализация по 14 темам на английском языке

На Рисунке 3.7 Можно наблюдать следующее: Темы 6, 8, 10, 11, 13, 14 связаны близко друг к другу. Это так же можно наблюдать по ключевым слова, к которым отнес темы алгоритм LDA. Например, к теме 2, относятся такие слова как: Регистрация, информация. К теме 7: Учебный, программа, а к теме 3: Технология, образование, школы. Таким образом все три темы были объединены в одну общую тему – Текущая учеба.

Автором экспериментальным путем было подобрано число непересекающихся тем для постов на английском и русском языке – 6 и 7 соответственно которые изображены на Рисунках 3.8 и 3.9.

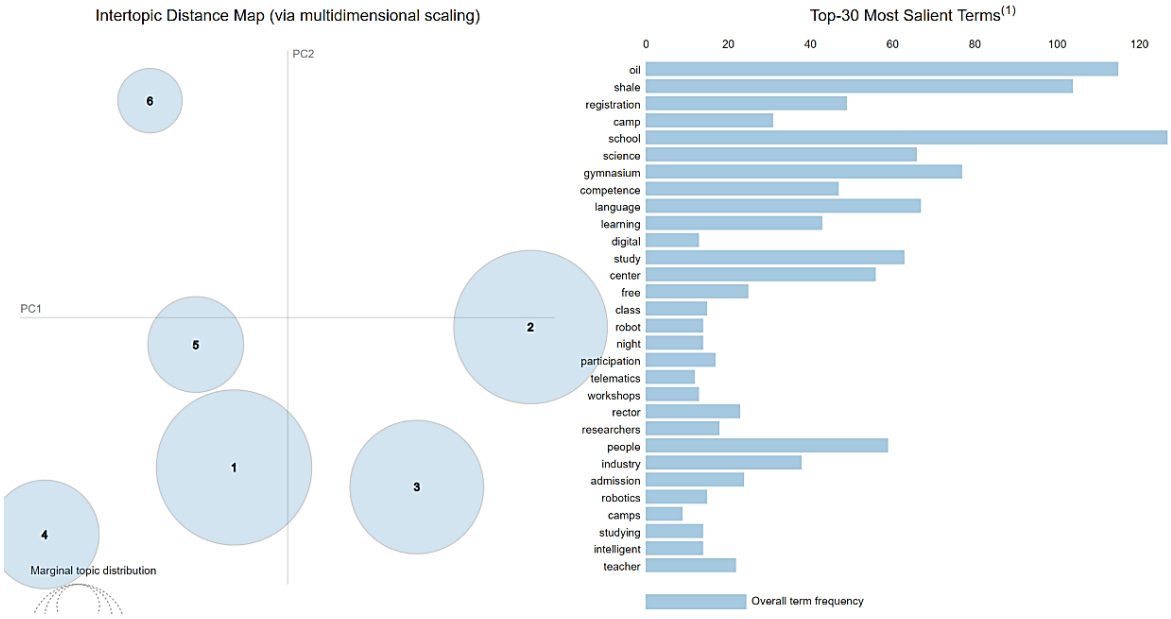

Рисунок 3.8 Распределение 6 тем и важных слов на английском языке

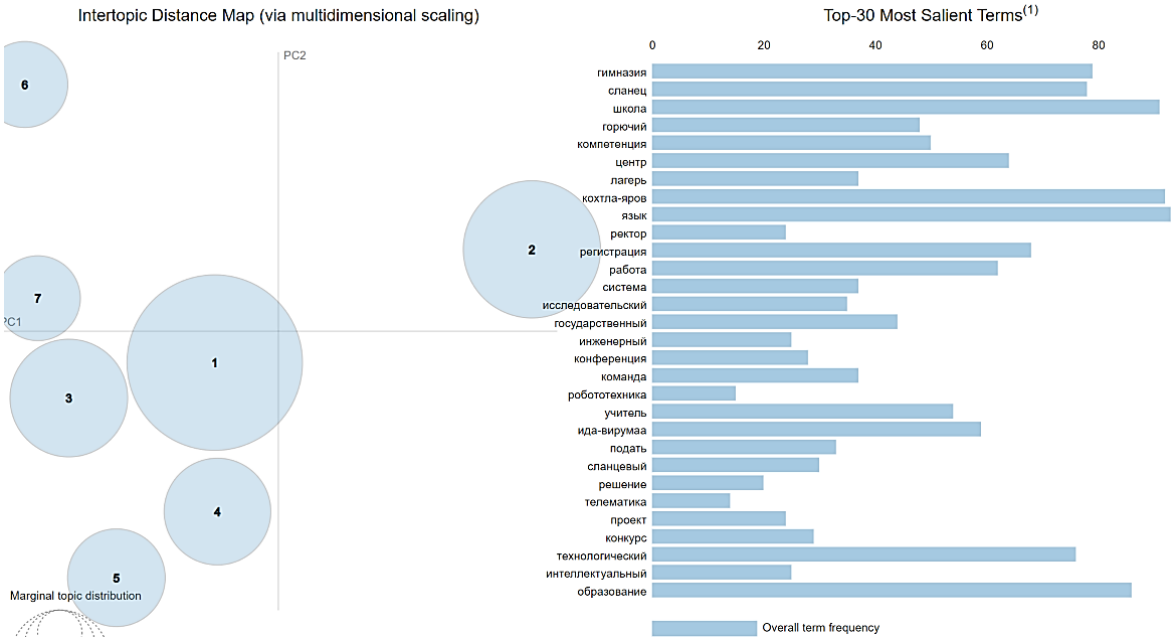

Рисунок 3.9 Распределение 7 тем и важных слов на русском языке

3.1.3 Запуск алгоритма LDA

На Рисунке 3.10 представлен запуск функции LdaModel, которая реализует алгоритм LDA. На вход подается объект-корпус с текстами, число тем, ранее подготовленный словарь со словами, число проходов алгоритма и параметры для более тонкой настройки алгоритма.

```
ldamodel = gensim.models.ldamodel.LdaModel (
    corpus, num_topics=NUM_TOPICS, id2word=dictionary,
    passes=10, random_state = rnd, update_every=1,
    alpha='asymmetric', eta='symmetric')
```

Рисунок 3.10 Запуск алгоритма LDA

В результате работы функции LdaModel будут сформированы наборы ключевых слов для каждой темы. Кроме того, каждому посту будет сопоставлен номер темы. Для интерпретации результатов необходимо вручную сформировать названия тем по спискам ключевых слов. Далее вместо номера темы каждому посту сопоставить сформированное название темы. Стоит отметить, что выявленные алгоритмом LDA темы могут повторяться. Ниже показан результат работы алгоритма касаемо выявления тем для постов на русском языке, при их числе равном 7.

(0, '0.017"технология" + 0.012*"обучение" + 0.010*"кохтла-яров" + 0.008*"образование"')*
(1, '0.009"информация" + 0.008*"регистрация" + 0.007*"язык" + 0.007*"робот"')*
(2, '0.012"гимназия" + 0.011*"школа" + 0.010*"язык" + 0.008*"государственный"')*
(3, '0.011"работа" + 0.009*"кохтла-ярве" + 0.008*"гимназия" + 0.007*"получить"')*
(4, '0.023"сланец" + 0.015*"центр" + 0.013*"компетенция" + 0.013*"горючий"')*
(5, '0.016"язык" + 0.009*"курс" + 0.008*"образование" + 0.006*"обучение"')*
(6, '0.012"школа" + 0.010*"лагерь" + 0.010*"исследовательский" + 0.009*"гимназия"')*

Перед программистом, использующим LDA, стоит задача присвоения названий для каждой найденной темы. Стоит отметить, что иногда эта задача может быть достаточно сложной в виду неоднозначности ключевых слов, которые выделил алгоритм. Для приведенного выше примера тематики были определены следующим образом:

0 – «текущая_учеба»
1 – «текущая_учеба»
2 – «школа_гимназия»
3 – «работа»
4 – «центр компетенции по горючему сланцу»
5 – «курсы и трененги»
6 - «школа_гимназия_лагерь»

Для английского языка было распределено 6 тем.

(0, '0.012"oil" + 0.010*"shale" + 0.007*"industry" + 0.006*"team"')*
(1, '0.024"school" + 0.015*"registration" + 0.013*"camp" + 0.013*"energy"')*
(2, '0.008"education" + 0.007*"study" + 0.007*"gymnasium" + 0.007*"people"')*
(3, '0.021"oil" + 0.019*"shale" + 0.011*"engineering" + 0.011*"science"')*
(4, '0.021"language" + 0.013*"study" + 0.012*"teacher" + 0.008*"learning"')*

(5, '0.035"school" + 0.028*"gymnasium" + 0.007*"competition" + 0.007*"education"')*

0 – «центр компетенции по горючему сланцу»
1 – «курсы_тренинги_лагерь»
2 – «текущая_учеба»
3 – «центр компетенции по горючему сланцу»
4 – «текущая_учеба»
5 – «текущая_учеба»

На финальном этапе работы алгоритм подготовит файл, где каждому посту будет сопоставлен номер темы. Фрагмент такого файла приведен ниже, перед каждым постом проставлен идентификатор его тематики.

6:Сегодня Международный день мытья рук!
3:Мы до сих пор отмечали День учителя в колледже, наслаждаясь прекрасной осенней погодой.
2:Спасибо всем, кто принял участие в праздновании дня рождения Вирумааского колледжа!

Далее необходимо в ручном режиме проверить, насколько правильно эти темы были сопоставлены постам.

3.2 Алгоритм со словарем и правилами

Законченных решений для определения тематик постов по словарю и правилам в открытых источниках автором не было найдено. На форумах, посвященных алгоритмизации, а также на страницах ресурса GitHub можно встретить словесные описания некоторых подходов к определению тематики текста по словарю. В целом алгоритм определения тематики будет сильно зависеть от конкретной предметной области, а именно от:

- количества возможных тематик;
- близости тематик по смыслу.

Предварительным шагом для реализации алгоритма определения тематики выгруженных постов будет подготовка словаря ключевых слов, а также набора правил для управления ходом работы алгоритма.

3.2.1 Подготовка словаря ключевых слов и правил

Для подготовки словаря и набора правил для определения тематики текста были использованы посты официальной страницы Вирумааского колледжа Таллиннского Технического Университета за 2021 год (данные за 2021 год будем считать обучающими). Словарь подготавливался исключительно в ручном режиме путем просмотра содержимого постов, анализа их тематики и извлечения ключевых слов,

характерных для той или иной тематики. Так, например, для тематики «Происшествия» был подготовлен следующий словарь ключевых слов:

["опасный","чрезвычайный", "ЧС","ЧП","болезнь"]

В ходе ручного анализа содержимого постов было отмечено, что ключевыми словами зачастую являются не отдельные слова, а словосочетания из двух слов. В связи с этим было принято решение о добавление таких словосочетаний в словарь в некотором удобном для последующей обработки вида. Словарь для тематики «Происшествия» принял следующий вид:

["опасный","чрезвычайный", "ЧС","ЧП", "дистанционный+обучение", "электронный+обучение", "болезнь", "носить+маска", "гибридный+обучение", "беречь+себя"]

Дальнейший анализ содержимого постов показал, что существуют такие ключевые слова, которые явно указывают, что текст к определенной тематике НЕ относится. Эта особенность была оформлена в виде правила: создать для каждой тематики два набора ключевых слов – один положительный (то есть индикаторы принадлежности текста конкретной тематике), другой – отрицательный (то есть индикаторы, что текст не принадлежит конкретной тематике). Ниже приведен пример полного словаря (с правилами) для тематики *«Курсы/тренинги/мероприятия»:*

{

"stopwords":["знакомиться","первый+курс","второй+курс","третий+курс","старший+курс","1+курс","2+курс","3+курс","лагерь","день+рождение","праздник","празднование","праздничный"],

"keywords":["курс", "онлайн-курс", "тренинг", "бизнес-курс", "мастер-класс","приглашать","приглашаться","приходить","мероприятие","участвовать","участие+бесплатное","конкурс","идея","маркетинг","регистрация","зарегистрироваться","стартер","игра","клуб","спортивный","баскетбол","баскетбольный","онлайн-мероприятие","конференция","викторина","церемония","ночь+исследователь","научный+театр","пресс-конференция","выставка","ярмарка"]

}

Приготовленные словари ключевых слов и правил находиться в Приложение 2.

3.2.2 Алгоритм определения тематики постов

При ручном визуальном анализе постов за 2021 год было выявлено, что во многих из них присутствуют ключевые слова, однозначно определяющие принадлежность поста к определенной тематике.

Например, в нескольких постах наблюдаются слова(словосочетания) «*ЧС», «ЧП», «дистанционное обучение».* Данные посты однозначно относятся к тематике «*Происшествия*» (какие-то из них об эпидемии коронавируса, какие-то о чрезвычайных ситуациях). Также можно отметить тематику «*Курсы/мероприятия/тренинги*», к которой можно однозначно отнести посты, по ключевым словам, "курс", "онлайн-курс", "тренинг", "бизнес-курс", "мастер-класс". Кроме того, имеется масса постов, не представляющих особой ценности для аналитики: посты о праздниках, поздравления кого-либо, краткие высказывания. Они будут отнесены к категории «*Прочее*».

Однако, существует набор постов, тематику которых идентифицировать, по ключевым словам, весьма проблематично. В первую очередь, это посты о текущей учебе и науке. И в той, и в другой тематиках достаточно много общих ключевых слов (например, «лаборатория», «исследование»).

В связи с этими выявленными особенностями было принято решение добавить правило построения алгоритма. Сначала пост будет проверяться на принадлежность тематикам, определить которые достаточно просто: *«Происшествия», «Прием на работу/учебу», «Курсы/Тренинги/Мероприятия»*. Перед проверкой на ключевые слова, которые относят пост к определенной тематике, он будет проверен на «стоп-слова», то есть слова, которые показывают, что пост к этой тематике не относится, и нужно перейти к проверке следующей тематики. Если пост не будет отнесен к первым трём тематикам, его анализ будет продолжен на принадлежность к оставшимся тематикам: *«Текущая учеба», «Наука и техника»*. В случае отсутствия ключевых слов по всем тематикам, пост будет отнесен к группе *«Прочее»*.

На Рисунке 3.11 представлена блок-схема алгоритма определения тематики поста.

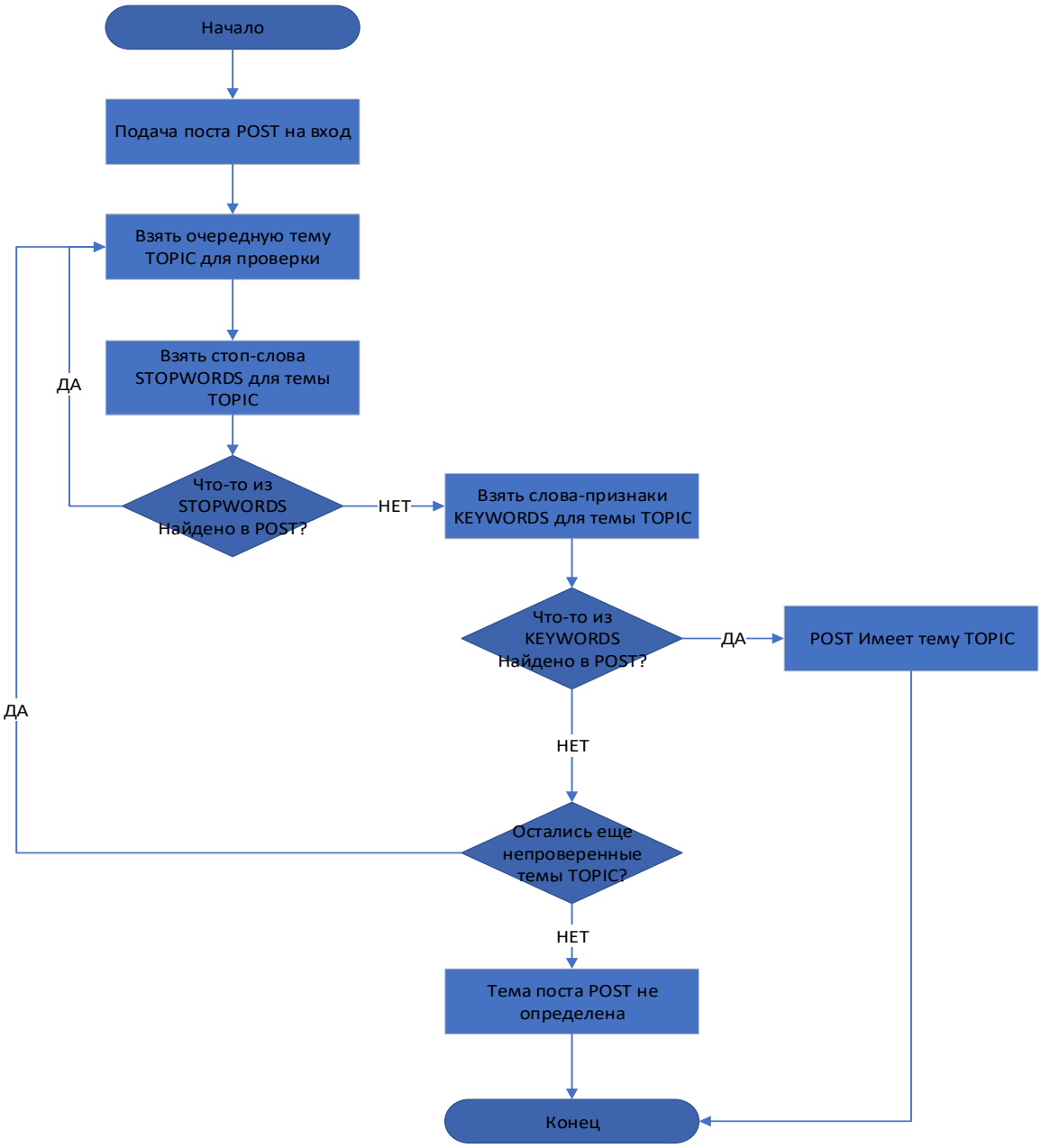

Рисунок 3.11 Блок-схема алгоритма определения тематики поста

3.3 Анализ полученных результатов

В данном разделе описывается анализ результатов работы алгоритмов по определению тематики постов. Для визуализации полученных результатов использовался модуль matplotlib, позволяющий строить диаграммы и выводить их на графической форме.

Для выполнения анализа результатов работы алгоритмов было предложено следующее: вручную просмотреть все посты в период с 2019 по 2020 годы и проверить, правильно ли определена тематика каждого поста. Стоит отметить, что

в данном случае будет иметь место субъективный фактор, поскольку разные люди могут считать, что один и тот же пост принадлежит к разным тематикам. В связи с этим было принято решение о проверке правильности работы алгоритма тремя проверяющими: автором настоящей выпускной работы и двумя независимыми лицами.

3.3.1 Распределение тематик в 2019 и 2020 годах

После определения проверяющими тематик всех выгруженных постов был проведен анализ количества постов тех или иных тематик за 2019 и 2020 год. На Рисунке 3.12 приведена диаграмма процентного соотношения постов по тематикам за 2019 и 2020 год.

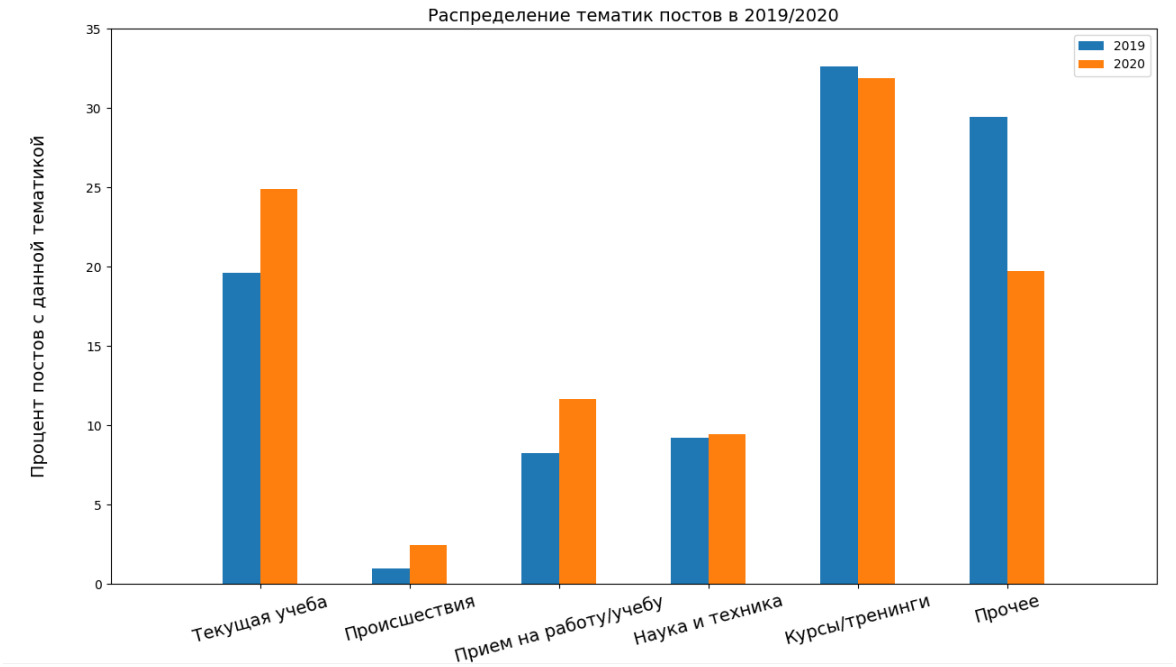

Рисунок 3.12 Процентное соотношение постов по тематикам за 2019 и 2020 год

Столбцы диаграммы показывают, какой процент от количества постов за весь год составляют посты конкретной тематики.

На диаграмме видно, что процент постов с тематикой «Происшествия» наименьший среди остальных.

Процент постов с тематикой «Текущая учеба», «Происшествия», «Прием на работу/учебу» за 2020 год вырос по сравнению с 2019 годом. Что касается «Наука и техника» и «Курсы/тренинги», то наблюдается незначительное различие количества постов в 2019 и 2020 году. Интересный результат можно наблюдать относительно тематики «Прочее». Согласно алгоритму, сюда были отнесены посты, содержащие ключевые слова: *"день рождения", "праздник", "празднование",*

"праздничный", "поздравление", "поздравлять", "юбилей". Кроме того, в тематику «Прочее» попадали посты, которые алгоритм не отнес к какой-либо другой тематике из ранее рассмотренных. По сравнению с 2019 годом, в 2020 году наблюдается существенное снижение таких постов (большая часть из них – это поздравления и какие-либо несущественные объявления).

3.3.2 Правильность определения тематик постов

3.3.2.1 Алгоритм LDA

Алгоритм LDA запускался для определения тематик постов на русском и английском языках. Для каждого языка было осуществлены запуски с количеством тематик, наиболее оптимальными для каждого языка. Эти оптимальные значения были вычислены ранее.

Ниже приведены результаты работы алгоритма для постов на русском и английском языках.

Посты на русском языке

Алгоритм в результате своей работы сформировал набор ключевых слов для каждой найденной темы. По этим ключевым словам, автор работы сформулировал названия тем. Ниже приведены названия выявленных тематик для количества тем N=7.

0 – «текущая_учеба»
1 – «текущая_учеба»
2 – «школа_гимназия»
3 – «работа»
4 – «центр компетенции по горючему сланцу»
5 – «курсы и трененги»
6 - «школа_гимназия_лагерь»

Отметим, что алгоритм выделил порядка семи различных тематик во всех постах за 2019-2020 годы, однако из-за схожих ключевых слов, их было выделено всего 5.

Далее необходимо проверить, насколько правильно алгоритм LDA классифицировал посты по найденным тематикам. Этот анализ был проведен вручную тремя проверяющими: автором настоящей работы и двумя независимыми лицами.

Ниже на Рисунке 3.13 приведена диаграмма, иллюстрирующие процент правильно определенных тематик для постов на русском языке при запуске алгоритма LDA для числа тем 7. На горизонтальной оси расположены тематики, которые выделил алгоритм LDA среди набора выгруженных постов. На вертикальной оси нанесена шкала процентов от 0 до 100. Каждый столбец показывает отношение числа

правильно определенных постов с определенной тематикой к общему числу постов с этой тематикой, то есть насколько правильно алгоритм LDA ставит соответствие между выгруженным постом и его тематикой.

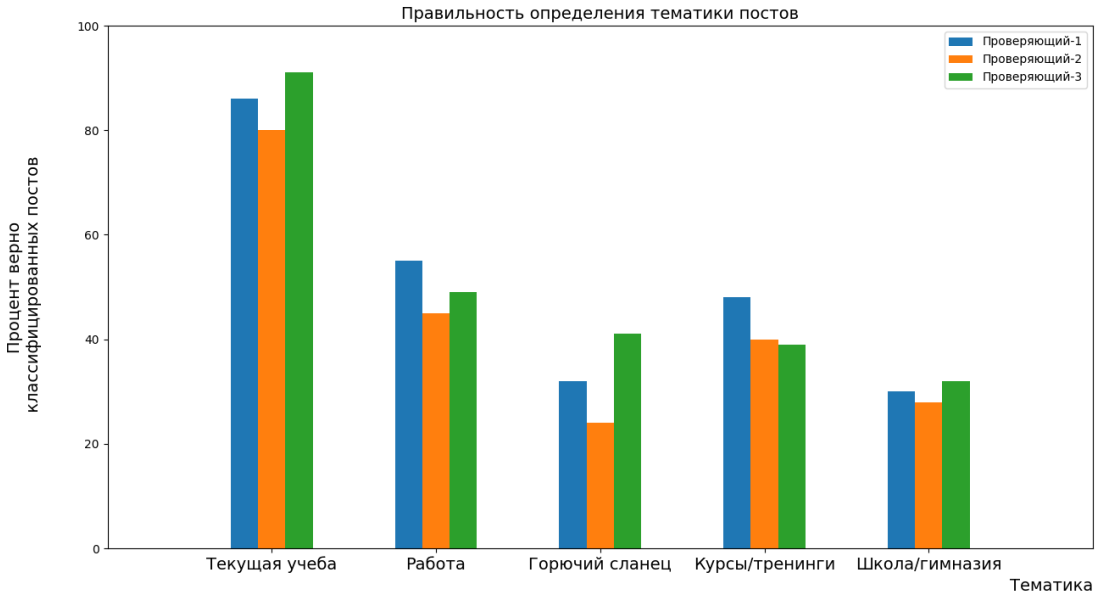

Рисунок 3.13 Правильность определения тематики постов (алгоритм LDA, 7 тем, RU)

Набольшую правильность определения тематики алгоритм показал для темы «Текущая учеба» (доля ошибочных определений составила менее 15%). Правильность определения остальных тематик составила порядка 45% и менее.

Полученные результаты нельзя признать удовлетворительными, так как доля ошибок весьма высока.

Посты на английском языке

Ниже приведены названия выявленных тематик для количества тем N=6.

0 – «центр компетенции по горючему сланцу»
1 – «курсы_тренинги_лагерь»
2 – «текущая_учеба»
3 – «центр компетенции по горючему сланцу»
4 – «текущая_учеба»
5 – «текущая_учеба»

Алгоритм LDA выделил 6 различных тематик во всех постах за 2019-2020 годы. Ниже на Рисунке 3.14 приведена диаграмма, иллюстрирующая процент правильно определенных тематик для постов на английском языке при запуске алгоритма LDA для числа тем 6, из-за схожих ключевых слов, их было оставлено всего 3. На горизонтальной оси расположены тематики, которые выделил алгоритм LDA среди набора выгруженных постов. На вертикальной оси нанесена шкала процентов от 0

до 100. Каждый столбец показывает отношение числа правильно определенных постов с определенной тематикой к общему числу постов с этой тематикой, то есть насколько правильно алгоритм LDA ставит соответствие между выгруженным постом и его тематикой.

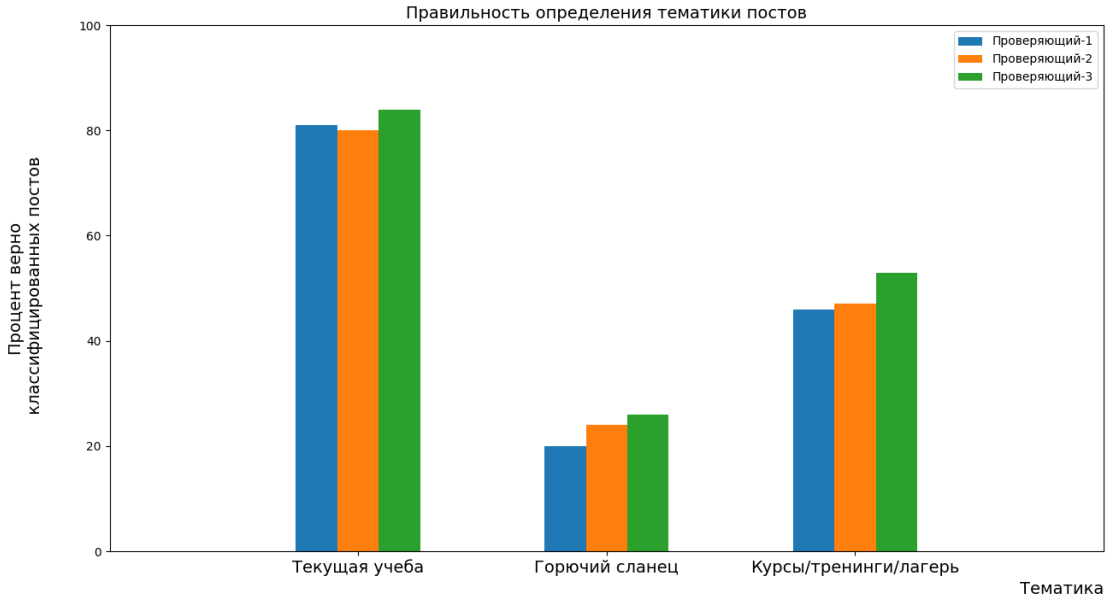

Рисунок 3.14 Правильность определения тематики постов (алгоритм LDA, 6 тем, ENG)

Набольшую правильность определения тематики алгоритм показал на тематике «Текущая учеба» (порядка 80%). Правильность определения остальных тематик составила порядка 50% и менее. Как и в случае запуска алгоритма LDA на постах на русском языке, полученная правильность слишком низка и не позволяет использовать результаты определения тематик постов для дальнейшего анализа.

3.3.2.2 Алгоритм по словарю и правилам

На Рисунке 3.15 изображена диаграмма, на которой указано, сколько постов в процентном соотношении определено правильно с помощью алгоритма с использованием словаря и правил.

Как можно видеть, алгоритм с использованием словаря и правил показал достаточно неплохие результаты: правильность определения тематик составила не менее 80%. То есть, к примеру, из 100 постов с тематикой «Происшествия» алгоритм правильно определил 80 постов с этой тематикой.

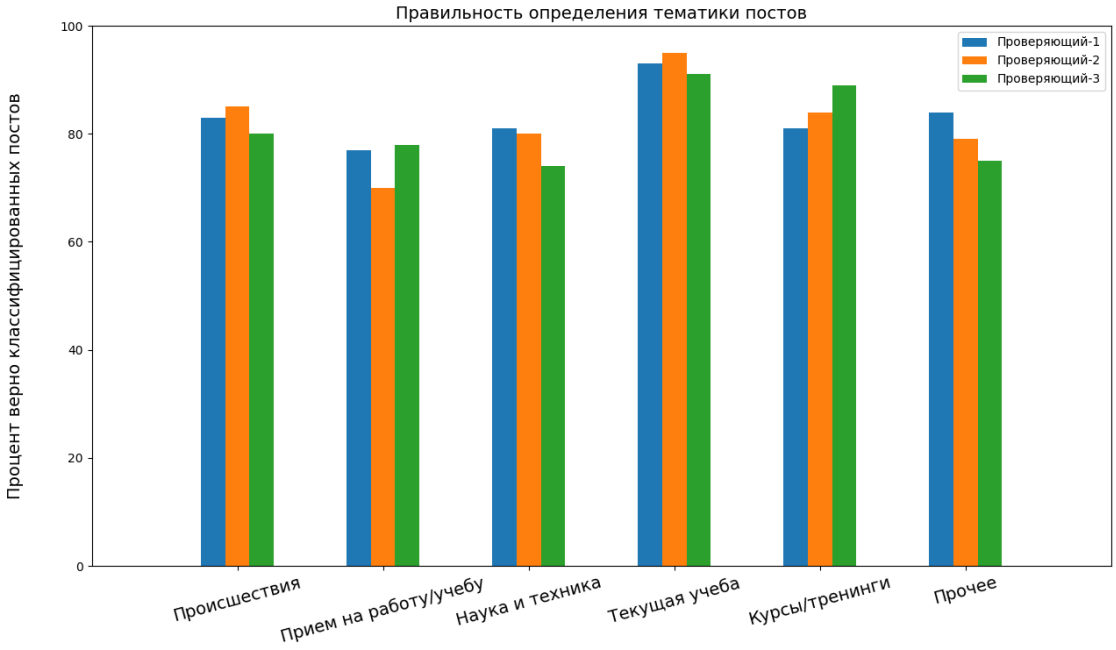

Рисунок 3.15 Правильность определения тематики постов для алгоритма по словарю и правилам (RU)

Полученные результаты признаны удовлетворительными для проведения дальнейшего анализа - в данной работе был далее выполнен анализ активности пользователей.

3.3.3 Аналитика активности пользователей

Под активностью будем понимать количество оставленных отметок «Мне нравится» и репостов записей. Также активность пользователей можно называть популярностью постов среди пользователей.

Активность пользователей вычислялась следующим образом. За определенный год для каждой тематики подсчитывалось число отметок «Мне нравится» и репостов. Затем это число делилось на число постов этой тематики за определенный год. Операция деления (нормирования) проводилась для того, чтобы получить не просто количество отметок «Мне нравится» и репостов, а некоторую средневзвешенную величину (можно называть ее рейтингом тематики). На Рисунке 3.16 показана диаграмма, иллюстрирующая популярность тематик среди пользователей за 2019 и 2020 год.

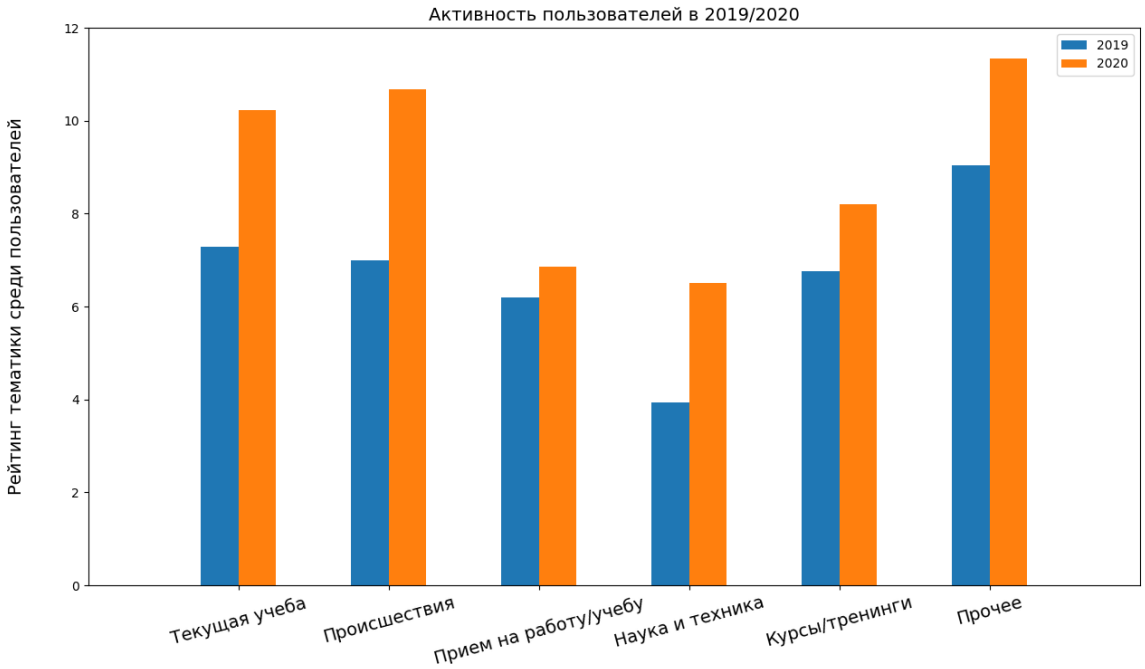

Рисунок 3.16 Популярность тематик среди пользователей за 2019 и 2020 год

В целом, на диаграмме видно, что активность пользователей заметно выросла в 2020 году по сравнению с 2019 годом. Можно видеть, что популярность тематик «Происшествия», «Наука и техника» и «Текущая учеба» выросла практически не оду треть в 2020 году по сравнению с 2019 годом.

Можно выдвинуть предположение, что активность пользователей по тематикам возросла в связи с увеличением времени нахождения их в онлайн. Кроме того, можно предположить, что пользователи посещали страницу Колледжа чаще, чтобы проверять появление новых новостей, связанных с организацией дистанционного обучения.

ЗАКЛЮЧЕНИЕ

В ходе написания данной выпускной работы были решены следующие задачи:

- Выбрана технология извлечения содержимого постов страницы социальной сети Facebook
- Выбрана и применена на практике технология машинного перевода текста
- Проведена предварительная обработка текстов для анализа
- Применены алгоритмы определения тематики поста
- Проведен сравнительный анализ результатов примененных алгоритмов.

Ниже кратко будут описаны основные этапы выполнения работы.

В качестве технологии извлечения содержимого постов со страниц социальной сети Facebook была применена технология Веб-скрейпинга, а именно модуль FacebookScrapper языка программирования Python. Применение этой технологии было обусловлено существенными ограничениями Facebook на использование Facebook API.

Машинный перевод текста был реализован с использованием платформы Google Translate. В связи с существенными ограничениями, накладываемыми Google на использование API от Google Translate, было принято решение отказаться от этого API. Дальнейший поиск методов использования Google Translate привел к технологии перевода, схожей с Веб-скрейпингом: средствами языка Python формирование запроса на сервис Google Translate, его отправка и извлечение результатов.

Для предварительной обработки текста использовался известный подход «Rule based POS tagging». Пост представлялся как набор входящих в него нормализованных слов, часть речи которых значима для анализа. Кроме того, порядок слов был сохранен для повышения точности алгоритма определения тематики.

В качестве алгоритмов определения тематики текста (классификации) использовался алгоритм LDA и был разработан алгоритм на основе словаря и правил. Стоит отметить, что данный алгоритм на основе словаря и правил существенно зависит от предметной области, то есть нет гарантии, что он подойдет для других предметных областей (определение тематик новостей, постов Twitter и прочих социальных сетей).

Результаты работы LDA были признаны неудовлетворительными в связи с большим количеством ошибок (более 50%) при определении тем. Алгоритм на основе

словаря и правил показал удовлетворительные результаты, процент ошибок не превысил 15–20%.

Итогом выполнения работы было проведение анализа полученных результатов определения тематик постов со страницы Вирумааского колледжа Таллинского технического университета. Были определены посты каких тематик преобладали в 2019 и 2020 годах. Кроме того, была проведена аналитика активности пользователей на странице Колледжа за 2019 и 2020 год. Соответствующие выводы по аналитике представлены в главе 3 данной выпускной работы.

Автор считает, что цель работы - применение методов сбора и анализа текстовых данных социальных сетей к странице Facebook Вирумааского колледжа - была достигнута.

По результатам проведенного анализа можно сделать вывод, что число постов в 2020 году выросло на 15%, по сравнению с 2019 годом (316 постов в 2019 году и 368 в 2020 году). Существенно выросла доля постов, посвященных текущей учебе и происшествиям. Активность пользователей (число отметок «Мне нравится» и репостов) выросла в 2020 году на 30% по сравнению с 2019 годом (2285 отметок и репостов в 2019 году и 3386 в 2020 году). Кроме того, в 2020 году существенно увеличился интерес пользователей к тематикам «Происшествия», «Наука и техника» и «Текущая учеба».

Что касается дальнейших направлений исследований, предлагается собрать статистику за 2021 год и проанализировать данные. Эти результаты позволили бы сделать выводы, растет ли популярность страницы колледжа с течением лет. Также представляет интерес информация о количестве человек, подписанных на страницу в 2019 и 2020 году, а также количество и частота посещений пользователями страницы колледжа. Кроме того, предлагается выполнить анализ постов на их оригинальном языке – эстонском.

KOKKUVÕTE

Selle lõputöö kirjutamise käigus on lahendatud järgmised ülesanded:

• Facebooki sotsiaalvõrgu postituste sisu väljavõtmise tehnoloogia valimine,

• teksti masintõlketehnoloogia valimine ja rakendamine,

• teksti eeltöötluse läbi viimine,

• postituse teema määramise algoritmide rakendamine,

• postituse teema määramise algoritmide töö tulemuste analüüs.

Töö põhietappe kirjeldatakse lühidalt allpool.

Facebooki sotsiaalvõrgu postituste sisu väljavõtmise tehnoloogiana kasutati veebilehe kraapimise tehnoloogiat, nimelt Pythoni programmeerimiskeele moodulit FacebookScrapper. Selle tehnoloogia kasutamine oli tingitud Facebooki API kasutamise piirangutest.

Teksti masintõlge viidi läbi Google Translate platvormi abil. Google'i rakenduse Google Translate API kasutamisele kehtestatud piirangute tõttu otsustati sellest API-st loobuda. Google'i Translate tõlkemeetodite kasutamine oli realiseeritud veebilehe kraapimisega sarnase tõlketehnoloogiga: Pythoni keele abil Google Translate teenuse taotluse vormindamine, selle saatmine ja tulemuste hankimine.

Teksti eeltöötlemisel kasutati tuntud lähenemisviisi: "Bag of words" ja "Rule based POS tagging". Postitus esitati selles sisalduvate normaliseeritud sõnade kogumina, mille kõneosad on analüüsi jaoks olulised. Lisaks teema tuvastamise algoritmi täpsuse parandamiseks postituses oli säilitud esialgne sõnade järjekord.

Postituse teema määramiseks kasutati *Latent Dirichlet Allocation* (LDA) algoritmi ja töötati välja ka märksõnade ja fraaside sõnastiku ning reeglitel põhineva algoritmi. LDA algoritmi töö tulemusi peeti mitterahuldavaks, kuna teemade määramisel oli palju vigu (üle 50%). Märksõnadel ja reeglitel põhinev algoritm andis rahuldavaid tulemusi, vigade protsent ei ületanud 15–20%. Tuleb ära märkida, et välja töötatud algoritm sõltub oluliselt teemavaldkonnast, see tähendab, et pole mingit garantiid, et see sobib ka teistele ainevaldkondadele (teemade määratlemine uudiste, Twitteri postituste ja muude sotsiaalvõrkude jaoks).

Töö tulemusena oli läbi viidud Tallinna Tehnikaülikooli Virumaa kolledži Facebooki postituste teemade määramise tulemuste analüüs. Tehti kindlaks, millised postitused millistel teemadel valitsesid aastatel 2019 ja 2020. Lisaks viidi läbi kolledži lehe aastatel

2019 ja 2020 kasutajate aktiivsuse analüüs. Vastavad järeldused on esitatud antud lõputöö 3. peatükis.

Autori arvamusel töö eesmärk - sotsiaalvõrkude tekstiandmete kogumise ja analüüsimise meetodite rakendamine Virumaa kolledži Facebooki lehele - on saavutatud.

Analüüsi tulemuste põhjal selgus, et postituste arv kasvas 2020. aastal 15% võrra võrreldes 2019. aastaga (316 postitust 2019. aastal ja 368 postitust 2020. aastal). On oluliselt kasvanud õppetööga ja juhtumustega seotud postituste osakaal. Kasutajate aktiivsus ehk meeldimiste (*like*) ja edasipostituste (*repost*) arv kasvas 2020. aastal 30% võrra võrreldes 2019. aastaga (2285 meeldimist ja edasipostitust 2019. aastal ja vastavalt 3386 2020. aastal). Lisaks sellele 2020. aastal on oluliselt kasvanud kasutajate huvi teemade "Juhtumused", "Teadus ja tehnoloogia" ning "Jooksev õppetöö" vastu.

Uurimuse edasiste sammudena oleks huvitav koguda ja analüüsida kogu 2021. aasta postituste andmeid. Need tulemused võimaldaksid järeldada, kas kolledži lehe populaarsus kasvab aastatega. Lisaks pakub huvi ka lehe jälgijate arv, samuti kasutajate külastuste arv ja sagedus kolledži lehel. Lisaks on plaanis analüüsida postitusi nende originaalkeeles - eesti keeles.

SUMMARY

The purpose of this thesis is to address the following goals:

- To choose a method of extracting the content of social network Facebook's posts.
- To choose and apply the most suitable machine translation technology.
- Text pre-processing performing.
- To implement topic modelling on Facebook's posts.
- Comparative analysis of the results of the applied algorithms.

The main stages of the thesis are briefly described below.

The chosen method of Facebook's post content extraction is web scrapping, in particular Python's Facebook-Scraper module. The reason of choosing this method is determined by Facebook's policy restricting the availability of APIs used.

At first, the attempt was to base the analysis on translation of Facebook posts using Google Translate API, however due to technical restrictions of this method it was decided to switch to a different approach: to query the Google Translate API using the Python library *mtranslate*.

For a preliminary content processing the "Rule based POS tagging" method was applied. The Facebook post was represented by a set of "normalized words" that could be analyzed. Furthermore, the words order was preserved in order to maintain the topic determination algorithm accuracy.

For Facebook's posts topics determination (classification) the Latent Dirichlet Allocation (LDA) technique was used, and "dictionary and rules" algorithm was developed. Noteworthy, dictionary and rules algorithm has subject limitation, which is dictated by specifics of required topics (its not necessarily will keep the same of efficiency outside of chosen topics range).

LDA algorithm has revealed inadequate results, due to high error level (more than 50%) while interpreting the posts' topics. In contrast, dictionary and rules algorithm has shown much less error-prone results with 15-20% of incorrectly interpreted posts topics.

As the result of the analysis, it was defined that the number of posts in 2020 was around 15% higher comparing to the same results of 2019 (316 posts in 2019 against 368 posts in 2020). Sufficient increase in the number of posts devoted to study-related news and accidents. User activity (likes and reposts) has increased by 30% in 2020 comparing to

2019 (2285 likes/reposts in 2019 against 3386 in 2020). A substantial increase in user's interest in the topics of "Accidents", "Science and technology" and "Current Studies".

The aim of the thesis was fulfilled, the Python-based social networks data collecting and text analysis methods were applied on the example of TalTech Virumaa College Facebook's posts dataset and a review of results was made.

For further analysis, the collection of 2021 statistics could be considered, which will allow to determine whether the popularity of College's Facebook page is increasing over years. Noteworthy mentioning the number of College's Facebook page subscribers in 2019 and 2020, as well as the frequency of user visits. The final suggestion is to conduct an analysis on Estonian language for higher accuracy results.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. A Gentle Introduction to the Bag-of-Words Model. [*Online*] https://machinelearningmastery.com/gentle-introduction-bag-words-model/ (15.12.2021).
2. How Does Text Preprocessing In NLP Work? [*Online*] https://medium.com/predict/how-does-nlp-pre-processing-actually-work-8d097c179af1 (15.12.2021).
3. Подходы к классификации. [*Online*] https://habr.com/ru/post/149605/ (15.12.2021).
4. Латентное размещение Дирихле (LDA). [*Online*] https://lambda-it.ru/post/tematicheskoe-modelirovanie-v-deistvii-lda (15.12.2021).
5. Язык программирования Python. [*Online*] https://www.python.org/ (15.12.2021).
6. Интегрированная среда разработки Pycharm. [*Online*] https://www.jetbrains.com/ru-ru/pycharm/ (15.12.2021).
7. Библиотека NLTK. [*Online*] https://www.nltk.org/ (15.12.2021).
8. Библиотека Gensim. [*Online*] https://pypi.org/project/gensim/ (15.12.2021).
9. Библиотека Mathplotlib. [*Online*] https://github.com/matplotlib/matplotlib (15.12.2021).
10. Морфологический анализатор. [*Online*] https://pypi.org/project/pymorphy2-dicts/ (15.12.2021).
11. Модуль для открытия URL-адресов. [*Online*] https://webformyself.com/python-urllib-request-i-urlopen/ (15.12.2021).
12. Pycharm — интеграция HTML и CSS. [*Online*] https://coderlessons.com/tutorials/python-technologies/uznaite-pycharm/pycharm-integratsiia-html-i-css (15.12.2021).
13. Wordrcloud - библиотека для создания облако слов. [*Online*] https://www.datacamp.com/community/tutorials/wordcloud-python (15.12.2021).
14. Re – модуль для операций с регулярными выражениями. [*Online*] https://docs.python.org/3/library/re.html (15.12.2021).
15. Репозиторий с проектом машинного перевода текста. [*Online*] https://github.com/mouuff/mtranslate(15.12.2021).
16. Topic Modeling with Gensim (Python). [*Online*] https://www.machinelearningplus.com/nlp/topic-modeling-gensim-python/#17howtofindtheoptimalnumberoftopicsforlda (15.12.2021).

ПРИЛОЖЕНИЯ

Приложение 1. Исходные коды приложения

https://github.com/Daenstar/topic-classifier

Приложение 2. Словарь для алгоритма классификации


```
{
	"Other":{
		"stopwords":[],

	"keywords":["день+рождение","праздник","празднование","праздничный","поздравление","поздравлять","юбилей"]
	},
	"Происшествия": {
		"stopwords":[],
		"keywords":["опасный","чрезвычайный", "ЧС","ЧП", "дистанционный+обучение", "болезнь", "носить+маска", "гибридный+обучение", "беречь+себя", "самоизоляция"]},

	"Прием на учебу/работу":{
		"stopwords":[],
		"keywords":["заявка","заявление", "приём","прием", "поступление","поступать", "подавать+документ","стойка+регистрация","приходить+учиться"]},


	"Курсы/Тренинги/Мероприятия":{

	"stopwords":["знакомиться","первый+курс","второй+курс","третий+курс","старший+курс","1+курс","2+курс","3+курс","лагерь"],

		"keywords": ["курс", "онлайн-курс", "тренинг", "бизнес-курс", "мастер-класс","приглашать","приглашаться","приходить","мероприятие","участвовать","участие+бесплатное","конкурс","идея","маркетинг","регистрация","зарегистрироваться","стартер","игра","клуб","спортивный", "баскетбол","баскетбольный","онлайн-мероприятие","конференция","викторина","церемония","ночь+исследователь","научный+театр","пресс-конференция","выставка","ярмарка"]},

	"Текущая учеба":{
		"stopwords":["сланец","сланцевый","горючий+сланец"],

		"keywords":  ["стипендия", "лектор", "лекция","первокурсник", "магистр", "магистреский",
"магистратура","лагерь","олимпиада","предмет","семинар","стажировка","гимназия","школа","выпускник","студент","директор+колледж","учебный+план","изучение","аудитория","знание","экзамен","учёба", "урок","абитуриент", "стажер",
"вебинар","учебный+год","электронный+обучение","предзащита","предварительный+защита"]},

	"Наука и техника":{
		"stopwords":[],
		"keywords": ["горючий+сланец", "топливо", "водород", "энергия", "робот", "робототехника",
"лаборатория","химия","физика","программирование","окружающая+среда","сланец","сланцевый","исследование","промышленность","космос","ракета","спутник",
		"машиностроении","энергетика","энергетический"]}
```