МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

SATBAYEV UNIVERSITY

Институт автоматики и информационных технологий
Кафедра Программная инженерия

Тулетаева Айымжан Аскаровна

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

На соискание академической степени магистра

Название диссертации	Разработка системы анализа больших данных для обработки текстов естественного языка, представленных в Интернете
Направление подготовки	7M06101– Software Engineering

Научный руководитель
PhD, ассоциированный профессор
_____Еримбетова А.С.
«__» _____ 2023 г.

Оппонент
PhD, ассоциированный профессор
_____Самбетбаева М.А.
«__» _____ 2023 г.

Нормоконтроль
_____Ахмедиярова А. Т.
«__» _____ 2023 г.

ДОПУЩЕН К ЗАЩИТЕ
Заведующий кафедрой ПИ
Кандидат физико-математических наук, профессор
_____ Молдагулова А. Н.
«__» _____ 2023 г.

Алматы 2023

Институт автоматики и информационных технологий

Кафедра «Программной инженерии»

Специальность: 7М06101– Software Engineering

УТВЕРЖДАЮ
Заведующий кафедрой ПИ
Кандидат физико-математических наук, профессор
_____ Молдагулова А. Н.
«__ » _____ 2023 г.

ЗАДАНИЕ

на выполнение магистерской диссертации

Магистранту: Тулетаевой Айымжан Аскаровной

Тема: «Разработка системы анализа больших данных для обработки текстов естественного языка, представленных в Интернете»

Срок сдачи законченной диссертации «__» _____

Исходные данные к магистерской диссертации: Определена диссертационная работа по разработке модели определения тональности текста на русском и казахском языках.

Перечень подлежащих разработке в магистерской диссертации вопросов или краткое содержание магистерской диссертации: а) разработка алгоритма автоматизированной семантической обработки новостных заголовков и комментариев; б) воспроизведение генерации текста и тематического моделирования; в) изучение и разработка архитектуры системы для семантической обработки текстов, а также принципов семантического анализа текстов.

Рекомендуемая основная литература:

1) Кобозева И.М. Лингвистическая семантика. — М., 2009
2) Автоматическая обработка текстов на естественном языке и анализ данных: учеб. пособие / Большакова Е.И., Воронцов К.В., Ефремова Н.Э., Клышинский Э.С., Лукашевич Н.В., Сапин А.С. — М.: Изд-во НИУ ВШЭ, 2017
3) Г., П. Методы автоматизированной обработки текстов. – М.: ИПИ РАН, 2008

ГРАФИК

подготовки магистерской диссертации

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Раздел 1. Обзор теории по обработке естественного языка с применением анализа тональности	04.11.2022	Выполнено
Раздел 2. Методы для разработки системы анализа тональности	20.02.2023	Выполнено
Раздел 3. Практическая реализация	15.04.2023	Выполнено

Консультации по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант, (уч. степень, звание)	Сроки	Подпись
Нормоконтроль	Ахмедиярова Айнур Танатаровна, PhD, ассоциированный профессор		
Программное обеспечение	Мукажанов Нуржан Какенович, PhD		
Антиплагиат	Аубакиров Бакдаулет, Ассистент		

Дата выдачи задания "____"_____ 2023 г.

Заведующий кафедрой _____Молдагулова А. Н.

Научный руководитель _____ Еримбетова А.С.

Задание принял к исполнению магистрант _____Тулетаева А.А.

Дата "_____"_____ 2023 г.

АННОТАЦИЯ

Современный мир сильно зависит от обработки больших объемов данных, которые постоянно поступают из различных источников. Интернет является одним из наиболее важных источников таких данных, включая тексты естественного языка, которые необходимо анализировать для получения ценной информации. Разработка системы анализа больших данных для обработки текстов естественного языка, представленных в Интернете, может существенно упростить этот процесс и повысить эффективность работы. Одним из главных преимуществ системы анализа больших данных является возможность автоматического извлечения значимой информации из больших объемов данных.

Сервисы социальных сетей и аналитические платформы быстро растут. В основном каждый день происходит большое количество различных событий, а также возрастает роль инструментов мониторинга социальных сетей. Социальные сети широко используются для управления и продвижения брендов и различных услуг. Таким образом, большинство популярных платформ социальной аналитики нацелены на бизнес-цели, в то время как мониторинг различных социальных, экономических и политических проблем остается недостаточно представленным и не охваченным тщательными исследованиями. Более того, большинство из них ориентировано на ресурсоемкие языки, такие как английский, тогда как тексты и комментарии на других малоресурсных языках, таких как русский и казахский языки, представлены в социальных сетях недостаточно хорошо.

В магистерской диссертации представлено исследование современных технологий и подходов, которые используются для автоматической обработки текстовой информации. Также в работе будет показана разработка системы анализа тональности с использованием методов и алгоритмов обработки естественного языка.

АҢДАТПА

Қазіргі әлем әртүрлі көздерден үнемі келетін үлкен көлемдегі деректерді өңдеуге өте тәуелді. Ғаламтор осындай деректердің ең маңызды көздерінің бірі болып табылады, оның ішінде құнды ақпарат алу үшін талдау қажет табиғи тілдегі мәтіндер. Интернетте ұсынылған табиғи тілдегі мәтіндерді өңдеуге арналған үлкен деректерді талдау жүйесін дамыту бұл процесті айтарлықтай жеңілдетеді және жұмыс тиімділігін арттырады. Үлкен деректерді талдау жүйесінің негізгі артықшылықтарының бірі - үлкен көлемдегі деректерден автоматты түрде мағыналы ақпаратты алу мүмкіндігі.

Әлеуметтік медиа қызметтері мен аналитикалық платформалар қарқынды дамып келеді. Негізінде, күн сайын көптеген түрлі оқиғалар орын алады, сонымен қатар әлеуметтік желілерді бақылау құралдарының рөлі де артып келеді. Әлеуметтік желілер брендтер мен әртүрлі қызметтерді басқару және жылжыту үшін кеңінен қолданылады. Осылайша, танымал әлеуметтік аналитикалық платформалардың көпшілігі бизнес мақсаттарына бағытталған, ал әртүрлі әлеуметтік, экономикалық және саяси мәселелерді бақылау жеткілікті түрде көрсетілмеген және қатаң зерттеулермен қамтылмаған. Оның үстіне, олардың көпшілігі ағылшын тілі сияқты ресурсты көп қажет ететін тілдерге бағытталған, ал орыс және қазақ сияқты ресурсты көп қажет ететін басқа тілдердегі мәтіндер мен түсініктемелер әлеуметтік желілерде жақсы көрсетілмейді.

Магистрлік диссертация мәтіндік ақпаратты автоматты түрде өңдеу үшін қолданылатын заманауи технологиялар мен тәсілдерді зерттеуді ұсынады. Сондай-ақ, жұмыста табиғи тілді өңдеу әдістері мен алгоритмдерін пайдалана отырып, сезімді талдау жүйесінің дамуы көрсетіледі.

ANNOTATION

The modern world is highly dependent on the processing of large amounts of data that constantly comes from various sources. A big data analytics system designed for processing natural language texts found on the Internet can greatly streamline the process and enhance productivity. It offers several notable benefits, with automatic extraction of valuable information from massive data sets being a key advantage. By leveraging such a system, the task of analyzing internet-based textual data becomes significantly easier and more efficient.

Social media services and analytics platforms are growing rapidly. Basically, a large number of different events take place every day, and the role of social media monitoring tools is also increasing. Social networks are widely used to manage and promote brands and various services. Thus, most of the popular social analytics platforms are aimed at business goals, while monitoring various social, economic and political issues remains underrepresented and not covered by rigorous research. Moreover, most of them target resource-intensive languages such as English, while texts and comments in other resource-intensive languages such as Russian and Kazakh are not well represented on social media.

The master's thesis presents a study of modern technologies and approaches that are used for automatic processing of text information. The paper will also show the development of a sentiment analysis system using natural language processing methods and algorithms.

Содержание

ВВЕДЕНИЕ

Актуальность темы

В современном мире текст является самым важным типом информации в программировании и в большинстве электронных хранилищ [1]. Большие данные хранятся в текстовых базах, библиотеках и во всех персональных компьютерах. Учитывая тот факт, что в нынешнее время значительного увеличился объем текстовой информации и сложную структуру текстов на естественном языке (NL) анализ текста можно считать актуальной задачей. Обработка естественного языка (Natural Language Processing) является дисциплиной, которая исследует проблемы взаимодействия компьютеров и естественных языков. Основной ее целью можно считать увеличение качества машинного анализа и синтеза информаций на естественном языке. Под машинным анализом подразумевается способность компьютера выводить смысл из входных естественно-языковых сообщений, а под машинным синтезом — способность компьютера производить выходные сообщения и любую другую информацию на естественном языке. В качестве информаций рассматриваются устные (разговорная речь) и письменные (текст) сообщения. Анализ тональности (Sentiment analysis) - хорошо известное направление в области обработки естественного языка. На протяжении многих лет проводились исследования многочисленных методов, контрольных показателей и ресурсов анализа тотальности.

Человеческий язык разнообразен о сложен. Мы можем выражать свои мысли в письменном и устном виде. Для каждого языка существует уникальный ряд синтактических и грамматических правил. Хоть контролируемое и неконтролируемое обучение, глубокое обучение, в настоящее время свободно применяются для моделирования человеческого языка, существует необходимость в синтаксическом и семантическом понимании и экспертных познаниях в предметной области, которые необязательно находятся в данных подходах машинного обучения. NLP важен, так как он помогает устранить неоднозначность в языке и использует полезную числовую структуру к информации для многих дальнейших приложений, подобных как распознавание речи или анализ текста.

Появление, а также развитие всемирной паутины и использование текстовой информации значительно повлияло на научно-техническую сферу и информационные системы. Данную область анализа текста принято называть автоматической обработкой текста (Natural Language Processing). Эта область включает в себя идеи обрвботуи текста на естественном языке, используемые в большинстве прикладных и коммерческих отраслях. Расширение потребления лингвистики в программировании способствует увелиению проблем, которые решаются с применением изучения и извлечения знаний в смежных научных

областях. Вычислительная лингвистика является междисциплинарной областью, объединяющей лингвистику, математику, информатику (Computer Science) и искусственный интеллект (Artificial Intelligence). Вычислительная лингвистика продолжает использовать и адаптировать методы и инструменты, разработанные в различных научных дисциплинах. На сегодняшний день с большой скоростью развиваются системы семантического поиска с использованием баз данных на основе программ глобальных организаций - производителей оборудования для серверов БД, например, Microsoft, Oracle и другие. Они построены на базе многомерных хранилищ, откуда происходит извлечение данных для их дальнейшей обработки. Масштабные поисковые системы в Интернете (Google, Яндекс и другие) используют методы поиска текстов, «похожих» на данные, и расчета релевантности найденных документов исходному запросу. Масштабные поисковые системы в Интернете, такие как Google, Яндекс и другие, применяют методы поиска текстов, схожих с исходными данными, и оценивают релевантность найденных документов к поисковому запросу. Компания Google постоянно стремится повышать релевантность результатов на страницах поисковых систем с каждым обновлением своего алгоритма. В настоящее время Google использует обновленную поисковую систему, основанную на двунаправленном кодировщике представлений Google от Transformers (BERT), который применяет методы обработки естественного языка. BERT считается одним из важных изменений, которое непосредственно затрагивает каждый десятый поисковый запрос.

Важным качеством естественного языка можно назвать его вариативность. Вариативность представлена наличием огромного числа отдельных языков, объединяющиеся в группы со сходными признаками и сами соединяют диалекты и наречия. Системы и подсистемы языков также неограниченно вариативны, выражены во многих альтернативах языковых единиц и явлений, их сочетаемости. Вариативность позволяет естественному языку быть в непрерывном развитии. Данное развитие представлено диахронными, многознаменательными процессами изменений и присутствием взаимозаменяемых явлений в синхронии. Вариативность языковых средств способствует последующему развитию языка. Необходимо учитывать тот факт, что изменения в языке обусловлены заложенными в нем тенденциями развития.

Тема обработки естественного языка (NLP) имеет огромную актуальность в современном информационном обществе. Обработка естественного языка является одной из ключевых областей искусственного интеллекта, и ее значимость продолжает расти в силу развития технологий и повсеместного использования текстовых данных.

Существует несколько факторов, которые делают тему обработки естественного языка актуальной и важной. Во-первых, объем текстовых данных, генерируемых и доступных в сети, постоянно растет. От социальных медиа до новостных сайтов и электронной корреспонденции, большое количество информации сегодня представлено в текстовой форме. Обработка и анализ этих данных позволяют извлечь ценные знания, выявить тенденции, определить настроения общества и многое другое [2].

Во-вторых, применение NLP находит широкое применение во многих отраслях и сферах деятельности. От медицины и финансов до маркетинга и правоохранительных органов, NLP играет важную роль в автоматической обработке документов, классификации текстов, извлечении информации, машинном переводе, чат-ботах и многих других приложениях [3]. Эти технологии позволяют автоматизировать задачи обработки текста, улучшить качество и эффективность работы, сократить затраты времени и ресурсов.

В-третьих, развитие глубокого обучения и компьютерных ресурсов привело к значительному прогрессу в области NLP. Появление глубоких нейронных сетей, таких как рекуррентные нейронные сети (RNN), сверточные нейронные сети (CNN) и трансформеры, привнесло новые методы и модели, которые способны эффективно обрабатывать текстовые данные и достигать высоких результатов в различных задачах NLP.

Наконец, коммуникация и взаимодействие человека с машинами становятся все более естественными. Голосовые помощники, чат-боты, автоматический перевод и другие NLP-технологии улучшают пользовательский опыт, делая интерфейсы более удобными и понятными. Это открывает новые возможности для более эффективного использования информации и улучшения коммуникации в различных областях жизни. В целом, актуальность темы обработки естественного языка обусловлена ростом объема текстовых данных, широким спектром приложений, доступностью новых методов и моделей, а также стремлением обеспечить более естественное взаимодействие между человеком и машиной. В дальнейшем, с развитием технологий и увеличением доступности данных, можно ожидать еще более впечатляющих достижений в области обработки естественного языка.

Объект исследования: новостные порталы, социальные сети из которых будут извлечены данные в виде текста для дальнейшего анализа.

Цель работы: Изучение основных понятий предметной области, классификация лингвистического программного обеспечения. Изучение, разработка и научное и практическое обоснование алгоритмов и методов автоматизированной семантической обработки текстов и их внедрение в технологии обработки текстов в системе управления электронными архивами.

Обучиться как манипулировать текстом для языковых моделей. Воспроизвести генерацию текста и тематическое моделирование. Изучить основы машинного обучения через более сложные концепции.

Предметная область: Обработка естественного языка в иностранной литературе используется параллельно с компьютерной лингвистикой, которая в свою очередь изучает когнитивный язык человека, а именно умение читать, понимать звуки на слух и переводить текст на разные языки. Автоматическая обработка естественного языка является частью прикладной лингвистики, которая используется в научно-технических областях для разработки программ на основе анализа текста.

Компьютерная лингвистика может называться математической, которая связывает сематические компоненты для разработки лингвистических программ и технологий на базе анализа естественного языка. Данная область занимается

декомпозицией текста, а именно разбиением предложений на токены, которое делается на вводе, и на выходе производится список разделенных слов. Токен является основой в анализе текста, так как его считывает программа с помощью котором происходит дальнейшая обработка. Его можно представить в виде последовательности, состоящая из чисел, букв и символов. Процесс разбиения текста на токены называется токенизацией.

В научных литературах об обработки текста часто используются термин "токен", для обычных пользователей иными словами можно назвать "слово", хотя эти термины по значению имеют разный смысл. В языкознании слово это сочетание букв, которое имеет значение, в то время как в компьютерной лингвистике оно представлено в виде знаков и символов, которые читает программа. В области компьютерной лингвистики исследуются статистические особенности распределения токенов в тексте (например, слов или символов), и на их основе разрабатываются формулы взвешивания, которые позволяют определить наиболее статистически значимые термины.

Задачи исследования

- Проведение тщательного анализа существующих технологий и систем для обработки текстов на естественном языке (NL).
- Разработка методов и алгоритмов для семантического анализа текстов, таких как алгоритмы поиска и классификации, основанные на особенностях естественного языка.
- Исследование области семантической обработки текстов на естественном языке и логических основ, необходимых для создания системы, которая может хранить и извлекать знания о грамматиках естественных языков и предметных областях текста.
- Изучение и разработка архитектуры системы для семантической обработки текстов, а также принципов семантического анализа текстов.
- Реализация автоматизированной системы для семантического анализа текстов и изучение предложенных методов и алгоритмов.

Литературный обзор на обработку текста на естественном языке

В работе "Обработка естественного языка. Ежегодный обзор Информационные науки и технологии" Чоудхури Г. отмечает три основные проблемы, связанные с созданием компьютерных программ, способных понимать естественный язык. Первая проблема связана с мыслительным процессом, вторая - с представлением и значением языкового ввода, а третья - с мировым знанием. Таким образом, системы обработки естественного языка могут начинаться с анализа отдельных слов для определения их морфологической структуры, частей речи, значения и т.д. Затем анализ расширяется на уровень предложения, включая порядок слов, грамматику, значение всего предложения и т.д. Наконец, система может учитывать контекст и общую предметную область, где конкретное слово или предложение имеют определенное значение или коннотацию.

В дальнейшем автор отмечает, что область обработки текста на естественном языке является важной и широко классифицируется как

манипулирование текстами для извлечения знаний, автоматического индексирования и реферирования, а также создания текста в нужном формате. Эта область позволяет структурировать большие объемы текстовой информации с целью извлечения конкретной информации или получения структур знаний, которые могут быть использованы для определенных целей. Системы автоматической обработки текста принимают текст в качестве ввода и преобразуют его в другую форму вывода.

В работе автор отмечает, что манипулирование текстами для извлечения знаний, автоматического индексирования и реферирования, а также для создания текста в нужном формате, является важной областью исследований в области обработки текста на естественном языке (NLP). Эта область позволяет структурировать большие объемы текстовой информации с целью извлечения конкретной информации или получения структур знаний, которые могут быть использованы для определенных целей. Системы автоматической обработки текста принимают текст в определенной форме ввода и преобразуют его в другую форму вывода, соответствующую задаче или требованиям.

Автор также ссылается на Элизабет Д. Лидди, профессора информатики и бывшего декана факультета информационных исследований Сиракузского университета, указывая, что центральной задачей систем обработки текста на естественном языке является преобразование потенциально неоднозначных запросов и текстов на естественном языке в однозначные внутренние представления, на которых может быть выполнено сопоставление и поиск.

Процесс обработки текста на естественном языке может начинаться с морфологического анализа, который включает создание терминов для получения морфологических вариантов слов, использованных в запросах и документах. Затем следует лексическая и синтаксическая обработка, которая использует лексиконы для определения характеристик слов, определения их частей речи, распознавания слов и фраз, а также анализа предложений.

В работе "Practical Natural Language Processing by Computer" [5] Роберт Мур обсуждает ограничения современных технологий обработки естественного языка (Natural Language Processing, NLP) и проблемы, связанные с распознаванием и пониманием непрерывной речи.

Автор отмечает, что для эффективной обработки и анализа естественного языка ввод должен быть представлен в машиночитаемой форме. Терминал или клавиатура позволяют пользователям ясно определить границы слов и предложений, что облегчает обработку текстовых данных. Однако, при распознавании речевого ввода, особенно без явных пауз между словами, возникают сложности.

Мур отмечает, что в настоящее время возможно распознавать отдельные слова или короткие фразы из ограниченного словарного запаса. Однако расшифровка непрерывной речи, где слова сливаются друг с другом без четких пауз, представляет значительные трудности для систем обработки речи. Решение проблемы непрерывной речи является активной областью исследований в NLP. Ученые и инженеры стремятся улучшить алгоритмы распознавания речи и

разработать модели, способные более точно интерпретировать непрерывную речь.

1 Основная часть

1.1 Введение понятия естественного языка

Обработка естественного языка (NLP) – подмножество методов искусственного интеллекта, которое используется для сокращения разрыва в общении между компьютером и человеком. Он возник из идеи машинного перевода, которая возникла во время Второй мировой войны. Первоначальная идея заключалась в том, чтобы преобразовать один человеческий язык в другой человеческий язык, например, превратить русский язык в английский язык, используя мозг Компьютеров, но после этого возникла мысль о преобразовании человеческого языка в компьютерный язык и наоборот, так что общение с машиной стало легким.

Понимание естественного языка помогает машинам «читать» текст (или другой ввод, например речь), имитируя способность человека понимать естественный язык, такой как английский, испанский, китайский и другие [6]. Обработка естественного языка включает в себя как понимание естественного языка, так и генерацию естественного языка, которые имитируют способность человека создавать текст на естественном языке, например. обобщить информацию или принять участие в диалоге.

Как технология обработка естественного языка достигла совершеннолетия за последние десять лет, когда такие продукты, как Siri, Alexa и голосовой поиск Google, используют NLP для понимания и ответа на запросы пользователей. Сложные приложения для анализа текста также были разработаны в таких различных областях, как медицинские исследования, управление рисками, обслуживание клиентов, страхование (обнаружение мошенничества) и контекстная реклама.

Современные системы обработки естественного языка представляют собой мощные инструменты, способные анализировать огромные объемы текстовых данных без усталости и предвзятости. Они обладают способностью понимать сложные концепции в контексте и разрешать двусмысленности языка, позволяя извлекать ключевые факты, отношения и создавать краткую сводку информации. В условиях огромного количества неструктурированных данных, генерируемых каждый день, от электронных медицинских записей до социальных медиа сообщений, автоматизация обработки текстовых данных стала критически важной для эффективного анализа и использования этой информации.

Обработка естественного языка является активно развивающейся областью искусственного интеллекта, которая привела к созданию мощных

инструментов для обработки, понимания и генерации естественного языка компьютерами. За последние десять лет технологии NLP достигли совершеннолетия и стали неотъемлемой частью нашей повседневной жизни.

Одной из основных областей применения NLP является машинный перевод. Системы автоматического перевода, такие как Google Translate, DeepL и Yandex.Translate, используют алгоритмы NLP для преобразования текста из одного языка на другой. Это значительно упрощает коммуникацию между людьми, говорящими на разных языках, и открывает новые возможности для международного взаимодействия.

Помимо машинного перевода, NLP нашел применение в таких областях, как анализ тональности текста, где компьютеры могут автоматически определять эмоциональную окраску текста, анализ сентимента в социальных медиа, где можно выявлять общественное мнение о продуктах или событиях, и извлечение информации, позволяющее автоматически извлекать ключевые факты и отношения из текстовых источников.

С развитием глубокого обучения и нейронных сетей, NLP достиг новых высот в создании систем, способных генерировать естественный язык. Примером этого является синтез речи, где компьютеры могут производить речь, звучащую естественно и понятно. Это приводит к развитию голосовых помощников, таких как Siri, Google Assistant и Amazon Alexa, которые могут отвечать на вопросы пользователей и выполнять различные задачи.

Современные системы обработки естественного языка позволяют компьютерам понимать контекст, обрабатывать большие объемы текстовых данных и создавать краткую сводку информации. Они становятся незаменимыми инструментами для бизнеса, медицины, научных исследований, финансового сектора и многих других областей, где необходимо анализировать и использовать текстовую информацию. Предполагается, что развитие NLP будет продолжаться, открывая новые возможности для более глубокого понимания и использования естественного языка компьютерами.

1.2 Компьютерная лингвистика и естественный язык

1.2.1 Лингвистический язык

В области автоматической обработки текстов на естественном языке было предложено множество перспективных идей, которые находят свою реализацию в различных прикладных системах, включая коммерческие продукты [7]. Компьютерная лингвистика продолжает активно расширять свою сферу применения, решая все новые задачи с помощью результатов смежных научных областей.

Компьютерная лингвистика — область, объединяющая знания из лингвистики, математики, компьютерных наук и искусственного интеллекта [8].

Он предполагает использование и адаптацию методов и инструментов, разработанных в этих науках, для достижения поставленных целей. Истоки компьютерной лингвистики можно найти в работах Н. Хомского по формализации структуры естественного языка, а также в ранних экспериментах программистов и математиков по машинному переводу, а также в первых разработках программного обеспечения в область искусственного интеллекта, направленная на понимание естественного языка.

Основным объектом обработки в компьютерной лингвистике являются тексты на естественном языке. Для развития компьютерной лингвистики необходимы базовые знания в области общей лингвистики. Лингвистика изучает общие закономерности естественного языка, его структуру и функционирование. Сюда входят такие области, как фонология (изучение звуков речи и правил их сочетания), морфология (строение и форма слов), синтаксис (строение предложения и правила порядка слов), семантика и прагматика (знание порядка слов). изучение значения слов). слова и предложения, а также контекстуальные особенности их употребления), а также лексикография (описание лексики того или иного естественного языка и приемы создания словарей).

Исследования в области компьютерной лингвистики продолжают привлекать внимание ученых и специалистов, так как они играют ключевую роль в разработке инновационных методов и технологий для обработки естественного языка. Ученые изучают различные аспекты семантической обработки текстов, включая алгоритмы классификации, поиска и анализа на базе естественного языка. Они стремятся разработать системы, которые способны понимать контекст и расшифровывать двусмысленность языка, чтобы извлекать ключевые факты и отношения или предоставлять резюме.

Применение компьютерной лингвистики охватывает множество областей. Например, в медицине, системы обработки естественного языка используются для автоматического анализа медицинских записей и справок, что помогает улучшить диагностику и лечение пациентов. В сфере финансов, анализ текстов позволяет проводить автоматическое мониторинговые новостей и социальных медиа для прогнозирования рыночных трендов и принятия инвестиционных решений. В области сетевого маркетинга и социальных сетей, анализ текстовых данных позволяет определять предпочтения и настроения пользователей, что помогает компаниям принимать целенаправленные маркетинговые меры.

Примеры применения компьютерной лингвистики также включают автоматический перевод текстов, создание виртуальных ассистентов и чат-ботов для общения с пользователями, автоматическую обработку больших объемов текстовых данных в сфере научных исследований и многое другое. В результате исследований в области компьютерной лингвистики постоянно расширяются возможности обработки и понимания естественного языка, что открывает новые перспективы для применения в различных отраслях и сферах деятельности.

1.2.2 Компьютерные системы компьютерной лингвистики

Компьютерная лингвистика постоянно расширяет свое применение, и ниже приведены некоторые известные прикладные задачи, которые решаются с использованием ее инструментов [9].

Одной из самых известных и первых задач компьютерной лингвистики является машинный перевод. Первые программы перевода появились в середине прошлого века, но быстро стало очевидно, что для достижения высокого качества перевода необходима более полная лингвистическая модель. В настоящее время существует широкий спектр компьютерных систем машинного перевода - от исследовательских проектов до коммерческих автоматических переводчиков. Заслуживают внимания также проекты многоязыкового перевода с использованием промежуточного языка, а также статистическая трансляция, основанная на анализе статистики переводных пар слов и словосочетаний. Хотя качество машинного перевода все еще далеко от совершенства, и исследования в области машинного обучения и нейронных сетей сыграли важную роль в его улучшении.

Еще одним важным приложением компьютерной лингвистики является информационный поиск и связанные с ним задачи индексирования, реферирования, классификации и рубрицирования документов [10]. Это включает разработку методов для эффективного поиска и извлечения информации из больших объемов текстовых данных. Компьютерная лингвистика помогает создавать интеллектуальные системы, которые могут обработать и организовать информацию, чтобы пользователи могли быстро найти нужные им документы и ресурсы.

Исследования в области компьютерной лингвистики продолжаются, и новые приложения и методы продолжают появляться. Они играют важную роль в автоматизации и улучшении обработки текстового материала, что приводит к развитию более эффективных инструментов для работы с естественным языком и повышению качества коммуникации и понимания между людьми и компьютерными системами.

Для осуществления полнотекстового поиска в больших базах текстовых документов требуется произвести индексирование текстов с использованием лингвистической предобработки и создания специальных индексных структур [11]. Векторная модель является наиболее известной и широко применяемой моделью информационного поиска. При этой модели информационный запрос представляется в виде набора слов, а подходящие документы определяются на основе схожести между запросом и вектором слов в документе. Другими словами, векторная модель использует сходство между запросом и документами для определения их релевантности.

Современные интернет-поисковые системы реализуют векторную модель, осуществляя индексирование текстов по употребляемым словам и применяя сложные процедуры ранжирования для предоставления релевантных документов. Однако активным направлением исследований в области информационного поиска является многоязыковой поиск по документам. Исследования в этой области направлены на разработку методов и алгоритмов, позволяющих эффективно и точно находить релевантные документы в многокультурной и многоязыковой среде.

Развитие информационного поиска продолжается, исследователи стремятся улучшить качество поисковых систем, сделать их более точными и эффективными в обработке текстовых документов. Многоязыковой поиск становится все более актуальным, учитывая разнообразие языков и культур в современном информационном пространстве.

При обработке естественного языка для ускорения поиска информации больших объемов данных используется реферирование текста, которое позволяет сократить объем данных и сокращает содержание [12]. Рефераты могут быть составлены как для отдельных документов, так и для группы связанных по теме материалов, например, для кластера новостных статей. Основной метод автоматического реферирования заключается в выборе наиболее значимых предложений из исходного текста на основе статистических показателей, таких как слова, словосочетания, а также структурные и лингвистические характеристики текстов.

При обработке больших коллекций документов классификация и кластеризация текстов имеют значительное значение. Классификация позволяет присваивать каждому документу определенную метку или категорию, основанную на заранее известных параметрах. Это позволяет организовать эффективный поиск и фильтрацию документов по заданным критериям. Кластеризация, с другой стороны, позволяет группировать текстовые документы, которые тематически близки друг другу, даже если их категории заранее неизвестны. Это может помочь в обнаружении скрытых паттернов и тематических структур в коллекции документов.

Text Mining, который является частью Data Mining, фокусируется на извлечении полезной информации из текстовых данных. Кроме классификации и кластеризации, Text Mining также включает в себя задачи извлечения ключевых слов, извлечения информации, суммаризации текста и анализа тональности. Эти методы находят широкое применение в различных областях, включая анализ социальных медиа, маркетинговые исследования, юридический анализ и многое другое.

Извлечение информации из текстов является незаменимои и прикладной задачей, связанной с Text Mining [13]. Она используется в экономическом и производственном анализе для выделения именованных сущностей, таких как имена персон, географические названия, названия фирм, а также для определения их отношений и связанных событий на основе частичного синтаксического анализа текста. Это позволяет обрабатывать большие объемы текстов, включая новостные потоки от информационных агентств. Выделенная

информация может быть структурирована и визуализирована для дальнейшего анализа и принятия решений.

Кроме того, в рамках Text Mining также решаются другие задачи, такие как выделение мнений и анализ тональности текстов [14]. Выделение мнений включает поиск и анализ мнений пользователей о товарах и других объектах, например, в блогах, форумах или интернет-магазинах. Анализ тональности текстов связан с оценкой общей тональности высказываний и текста в целом, аналогично классическому контент-анализу текстов массовой коммуникации. Эти задачи имеют широкое применение в маркетинге, социальном мониторинге и других областях, где важно понимать общественное мнение и эмоциональную окраску в текстовых данных.

1.3 Анализ предложения

Цель анализа предложений состоит в том, чтобы определить, что означает предложение. В компьютерных терминах каждому входному предложению должно быть присвоено какое-то смысловое представление. В предложении сочетаются два типа информации, которые позволяют говорящим достичь интерпретации этого предложения: его синтаксическая структура и лексическое значение слов, из которых оно состоит. Таким образом, морфология слов важна для синтаксического анализа. Как только структура предложения определена, семантика отвечает за то, как слова и структуры объединяются для выражения значения.

1.3.1 Морфологический анализ

Предложения состоят из слов. Поэтому важным шагом в процессе синтаксического анализа является поиск слов в словаре. Этап поиска не является тривиальным [15]. Когда слова появляются в предложениях, они обычно имеют флективные окончания. Словообразовательные аффиксы также очень распространены и часто изменяют часть речи корня, к которому они присоединяются. И мы должны иметь в виду, что добавление аффиксов может повлечь за собой изменение правописания. Вот несколько примеров из английского языка, чтобы проиллюстрировать, с какими проблемами сталкивается компьютерный лингвист, пытаясь формализовать все морфологические факты языка.

Рассмотрим спряжение английского глагола. Предположим, что предложение, которое мы пытаемся проанализировать, таково:

She always stops at a red light.

При поиске слова «stops» мы ожидаем получить такой анализ:

{stop, глагол} + {-S, 3-е лицо единственного числа, настоящее время}

Наш морфологический анализатор должен будет разделить последовательность «S t o p s» на две корректные морфы, соответствующие корню {stop} и флективному окончанию {-S} соответственно. Для того чтобы морфологический анализатор мог идентифицировать два. морфы, они должны быть в словаре, к которому обращается программа. А пока можно предположить, что в словаре есть как {stop, verb}, так и {-S, 3-е лицо единственного числа, настоящее время}. Кажется неважным, представляют ли элементы словаря морфемы (абстрактные единицы) или морфы (конкретные реализации), потому что представление морфем и представление морфов имеют точно такие же символы. Теперь предположим, что глагол не «stops», а «spies». В этом случае словарные статьи {spy, verb} и {-S, 3-е лицо единственного числа} могут никогда не быть идентифицированы анализатором, который имеет дело с последовательностью «s p i e s». Вероятно, первое решение, которое приходит на ум, состоит в том, чтобы иметь две записи для глагола «шпион»: {spy, глагол} и {spie, глагол}, с тегом, сигнализирующим, что это один и тот же глагол, имеют одинаковое значение и т. д. Это, безусловно, может быть единственным решением, если чередование написания «y / ie» влияет только на глагол «spy»: именно так и происходит морфологическая обусловленность (например, men во множественном числе от man, deer во множественном числе от deer). Однако, поскольку многие слова (существительные и глаголы) демонстрируют одинаковое изменение правописания, может быть более эффективным включить лингвистическое обобщение: все корни, оканчивающиеся на согласную + y, претерпевают изменение правописания с y на ie перед флексиями, которые не начинаются с i. Обобщения позволяют сэкономить место для хранения, что, в свою очередь, сокращает время, необходимое для поиска предметов.

Возможный подход к проблеме состоит в том, чтобы рассматривать изменение правописания как правило или инструкцию соотносить входные слова, оканчивающиеся на ie, в соответствующих условиях с элементами словаря, оканчивающимися на y. Таким образом, словарь должен включать только представления морфем (например, {spy, глагол}, но не каждой конкретной реализации {spy, глагол, не от третьего лица в единственном числе}, {spie, глагол, от третьего лица в единственном числе}. Программа анализа морфологии идентифицирует потенциальные морфы во входном слове. Если обнаружено возможное изменение написания (например, из y в последовательности «S p i e» в «S p i e S»), небольшая подпрограмма возвращает морфему-кандидата. {spy}, который затем проверяется в словаре. Такой подход называется двухуровневой морфологией, так как существуют «поверхностные» формы (входящие слова) и «лексические» формы (в словаре). два уровня достигается за счет небольших подпрограмм.

1.3.2 Анализ синтаксиса: парсеры и грамматики

С точки зрения логики или математики, наблюдение за синтаксическим анализом предложения эквивалентно выяснению того, принадлежит ли это предложение множеству возможных предложений языка, и, если это так, получению представления о его структуре, т.к. пример в виде синтаксического дерева или заключенных в скобки предложений с метками в качестве нижних индексов [16]. Как известно, количество предложений, возможных в языке, бесконечно. К счастью, членство в этом бесконечном множестве, по-видимому, можно охарактеризовать с помощью конечного набора правил. Такие характеристики языков называются «грамматиками». Вместе с лексикой (или словарем) грамматики можно считать знанием языка. В компьютерной лингвистике процесс синтаксического анализа называется «разбором». Что требуется компьютеру для разбора предложения определенного языка? Во-первых, компьютер должен владеть этим конкретным языком (то есть грамматикой и словарем). Во-вторых, компьютер должен уметь пользоваться знаниями языка. Нам нужна программа, которая принимает предложение в качестве входных данных и находит комбинацию грамматических правил и слов словаря, описывающих это предложение. Такие программы называются «парсерами».

Парсеры используют грамматики. Оба могут быть интегрированы по-разному или, скорее, в разной степени. На одном конце находятся синтаксические анализаторы, содержащие инструкции, эквивалентные грамматическим правилам. Разделение между грамматикой и парсером менее четкое. С другой стороны, существуют системы, в которых грамматика и синтаксический анализатор разделены как разные модули, так что один и тот же синтаксический анализатор можно использовать с грамматиками разных языков. Другое преимущество отделения синтаксического анализатора от грамматики заключается в том, что это экономит время и усилия автора грамматики (человека, который пишет правила, описывающие конкретный язык) и программиста, создающего синтаксический анализатор. Если синтаксический анализатор не принимает грамматическое предложение, проблема может заключаться в любом компоненте: грамматическая ошибка или синтаксический анализатор может работать неправильно.

С точки зрения логики и математики, количество возможных предложений в языке может быть бесконечным. Однако для анализа предложений их членство в этом бесконечном множестве можно охарактеризовать с помощью конечного набора правил. Эти правила составляют грамматику языка. Грамматика определяет, какие комбинации слов и фраз являются корректными предложениями, а какие нет. Грамматика включает в себя правила синтаксиса, которые определяют структуру предложений, и правила лексики, которые определяют, каким словам принадлежит язык.

Для проведения синтаксического анализа предложений компьютеру необходимо обладать знанием конкретного языка, то есть грамматикой и

словарем. Он должен уметь использовать это знание для анализа предложений. В этом помогают специальные программы, называемые парсерами.

Парсеры являются программами, которые принимают предложение в качестве входных данных и анализируют его, находя сочетания грамматических правил и слов словаря, которые описывают данное предложение. Парсеры используют грамматику и лексику языка, чтобы разобрать предложение на составляющие его части и определить их синтаксические отношения.

Существует несколько подходов к синтаксическому анализу. Один из них – это синтаксический анализ сверху вниз, при котором анализ начинается с главного предложения и разбивается на подпредложения и фразы до достижения листьев синтаксического дерева. Другой подход – это синтаксический анализ снизу вверх, который начинается с отдельных слов и комбинирует их в более крупные фразы и предложения.

Синтаксический анализ имеет широкое применение в различных областях, включая компьютерные языки программирования, машинный перевод, автоматическую обработку естественного языка и многое другое. Парсеры играют важную роль в этих областях, позволяя компьютерам понимать и обрабатывать человеческий язык. Этот процесс важен для обработки и понимания естественного языка компьютерами и находит применение в различных областях.

1.4 Сложности моделирования естественного языка

Моделирование в компьютерной лингвистике представляет собой сложный процесс из-за особенностей естественных языков. Естественные языки являются обширной системой знаков, состоящая из множества уровней, которая возникла для обмена информацией в процессе человеческой деятельности и постоянно изменяется в связи с этой деятельностью [17].

Текст на естественном языке состоит из отдельных единиц, и возможны различные способы разбиения текста на такие единицы, которые относятся к разным уровням языка. Это создает сложность при моделировании и анализе текста, поскольку необходимо учитывать разнообразие языковых структур и их взаимосвязи. Исследования в области компьютерной лингвистики направлены на разработку моделей и методов, способных эффективно управлять этой сложностью.

Это включает в себя создание формальных грамматик, лексических ресурсов, алгоритмов анализа и синтеза текста, а также применение методов машинного обучения для автоматического извлечения знаний из текстовых данных.

Основной вызов заключается в создании моделей, которые могут учитывать многозначность, контекстуальную зависимость и другие особенности естественного языка. Это требует учета разных уровней языка, таких как фонетика, морфология, синтаксис, семантика и прагматика, и их

взаимодействия. Сложность моделирования в компьютерной лингвистике обусловлена многоуровневой структурой естественных языков и неоднозначностью их текстового представления, требуя разработки специализированных методов и моделей для эффективной обработки и анализа текстовой информации.

В языковой лингвистике можно выделить следующие уровни анализа языка [18]:

1. Фонологический уровень языка связан с его звуковыми аспектами, которые представлены фонемами. Данный уровень особенно важен для устной речи, где происходит восприятие и произношение звукового материала. В письменных текстах, особенно в языках с алфавитной системой письма, этот уровень соответствует символам, то есть буквам алфавита, которые представляют звуки.

2. Морфологический уровень языка относится к анализу слов и их форм, известных как словоформы. На этом уровне рассматриваются различные морфологические элементы, такие как морфемы, которые являются минимальными значимыми частями слова. Морфемы могут включать корни, приставки, суффиксы, окончания и постфиксы, которые вместе образуют разнообразные словоформы.

3. Синтаксический уровень: это уровень предложений или высказываний. Здесь анализируется структура предложений, включая синтаксические связи между словами, порядок слов и их функции.

Кроме этих основных уровней, существуют также другие подуровни и подсистемы. Например, на морфологическом уровне можно выделить морфемы как более детальные компоненты слова.

Исследования в области лингвистики продолжаются, и количество уровней анализа языка, а также их перечень, являются предметом изучения. В дополнение к фонологическому и морфологическому уровням, некоторые ученые также выделяют лексический уровень. Лексический уровень охватывает лексемы, то есть все конкретные грамматические формы слова. Лексемы записываются в словарях естественных языков, где фиксируется каноническая форма слова, называемая леммой. Анализ языка включает различные уровни, которые представляют собой подсистемы общей системы естественного языка. Кроме того, каждый уровень может содержать в себе подуровни и подсистемы, которые могут быть выделены для более детального анализа.

На синтаксическом уровне анализа языка можно выделить несколько подуровней и надуровней.

На синтаксическом уровне анализа языка выделяются подуровни, связанные с словосочетаниями, а также надуровень сложного синтаксического целого, который включает последовательность предложений. Иерархия уровней обусловливает организацию и связь между единицами языка на разных уровнях.

Одним из важных аспектов является семантический уровень языка, где смысл присутствует в знаковых единицах, таких как морфемы, слова и предложения. Организация семантического уровня еще не полностью понята, но предполагается существование универсального набора элементарных

семантических единиц, называемых "семами", которых примерно 2 тысячи. С помощью этих сем можно выразить смысл любого высказывания.

Моделирование естественного языка усложняется не только многоуровневостью языковой системы, но и его постоянными изменениями. Язык меняется со временем, затрагивая не только словарный запас, но и синтаксис, морфологию и фонетику. Это делает невозможным создание единоразовой формальной модели для конкретного языка и соответствующего лингвистического процессора.

Одной из сложных задач при обработке текстов на естественном языке является неоднозначность его единиц, которая проявляется на всех уровнях и выражается в полисемии, омонимии и синонимии.

- Полисемия относится к наличию у одной лингвистической единицы нескольких связанных значений. Например, слово "вода" может означать "жидкость", "аква" или "влага".
- Омонимия включает совпадение по форме двух или более разных по смыслу единиц. Виды омонимии включают лексическую омонимию (слова, звучащие и пишущиеся одинаково, но имеющие разные значения), морфологическую омонимию (совпадение форм одной лексемы), лексико-морфологическую омонимию (совпадение словоформ разных лексем) и синтаксическую омонимию (неоднозначность синтаксической структуры, приводящая к нескольким возможным интерпретациям).

Таким образом, на синтаксическом уровне анализа языка выделяются подуровни, связанные с словосочетаниями, и надуровень сложного синтаксического целого, который включает последовательность предложений. Иерархия уровней обусловливает организацию и связь между единицами языка на разных уровнях.

Важным аспектом является семантический уровень языка. Смысл присутствует во всех знаковых единицах языка, таких как морфемы, слова и предложения. Значительным подтверждением независимости семантического уровня является то, что люди обычно запоминают смысл высказывания, а не его конкретную языковую форму. Организация семантического уровня до сих пор не полностью понята, но предполагается, что существует универсальный набор элементарных семантических единиц, называемых "семами", и их количество составляет приблизительно 2 тысячи. С помощью этих сем можно выразить смысл любого высказывания.

Кроме многоуровневости языковой системы, моделирование естественного языка также осложняется его постоянными изменениями. Со временем в языке происходят изменения, затрагивающие не только словарный запас (появление новых слов и новых значения для существующих), но также синтаксис, морфологию и фонетику. В результате невозможно разработать единоразовую формальную модель для конкретного естественного языка и создать соответствующий лингвистический процессор. Одной из наиболее сложных задач при обработке текстов на естественном языке является неоднозначность его единиц, которая проявляется на всех уровнях и выражается в явлениях полисемии, омонимии и синонимии.

Полисемия означает наличие у одной языковой единицы нескольких связанных значений. Например, слово "земля" может означать "сушу", "почву" или "конкретную планету".

Синонимия представляет собой полное или частичное совпадение значений разных единиц. Например, слова "негодяй" и "подлец" являются синонимами, а также приставки "пре-" и "пере-" (прекрасный, пересохший).

Омонимия включает совпадение по форме двух разных по смыслу единиц. Различают несколько видов омонимии, таких как лексическая омонимия (слова, которые звучат и пишутся одинаково, но не имеют общих значений), морфологическая омонимия (совпадение форм одной лексемы), лексико-морфологическая омонимия (совпадение словоформ разных лексем) и синтаксическая омонимия (неоднозначность синтаксической структуры, ведущая к нескольким возможным интерпретациям).

Таким образом, семантический уровень языка является особым и присутствует в знаковых единицах. Моделирование естественного языка сложно из-за изменений, которые происходят в нем, а также из-за неоднозначности его единиц, такой как полисемия, омонимия и синонимия.

1.5 Обработка естественного языка корпоративного уровня

Применение расширенной аналитики представляет собой реальную возможность в фармацевтической и мед отраслях, где проблема содержится в подборе подходящего решения, а затем его действенном внедрении на предприятии.

Для эффективной обработки естественного языка необходимо ряд функций, которые должны быть интегрированы в любое решение NLP корпоративного уровня, и некоторые из них описаны ниже [20].

Аналитические инструменты

Существует огромное разнообразие в составе документов и текстовом контексте, включая источники, формат, язык и грамматику. Для борьбы с этим разнообразием требуется ряд методологий [21]:

1. Преобразование форматов документов (например, HTML, Word, PowerPoint, Excel, текст PDF, изображение PDF) в стандартизированный формат с возможностью использования поиска;
2. Возможность идентификации, маркировки и поиска в конкретных разделах документа: фокусировка поиска на удалении шума из справочного раздела документа;
3. Лингвистическая обработка для определения значимых единиц в тексте, таких как предложения, категории существительных и глаголов, а также связей меж ними;
4. Семантические инструменты, которые идентифицируют понятия в тексте, таковые как лекарства и болезни, и нормализуют определения из

стандартных онтологий. В дополнение к основным онтологиям медико-биологической науки и здравоохранения, таким как MedDRA и MeSH, возможность прибавления своих словарей является непременным условием для многих организаций;

5. Распознавание образов для обнаружения и идентификации категорий информации, которые тяжело установить с помощью лексикографического подхода. К ним относятся даты, числовая информация, биомедицинские термины (например, концентрация, объем, дозировка, энергия) и мутации генов/белков;
6. Возможность обрабатывать интегрированные таблицы в тексте, независимо от того, отформатированы ли они с использованием HTML или XML или как свободный текст.

Расширенная аналитика имеет огромный потенциал и в других отраслях, помимо фармацевтической и медицинской. Вот некоторые примеры применения расширенной аналитики в различных областях.

1. Финансовая отрасль: В финансовом секторе аналитические инструменты могут использоваться для обработки и анализа больших объемов финансовых данных, прогнозирования трендов рынка, определения рисков и выявления мошеннической активности. Автоматическое распознавание и обработка текстов из финансовых отчетов, новостных статей и социальных медиа помогает в принятии обоснованных финансовых решений.
2. Розничная торговля: Расширенная аналитика может быть применена для анализа данных о покупателях, прогнозирования спроса, оптимизации ценообразования и управления запасами. Текстовые данные из обзоров товаров, отзывов покупателей и социальных медиа могут быть обработаны для выявления тенденций и предпочтений покупателей.
3. Страхование: В страховой отрасли расширенная аналитика может быть использована для автоматизации процессов подписки и оценки рисков, обработки и анализа исторических данных, выявления мошенничества и определения оптимальных тарифов. Обработка текстовых данных из заявлений о страховом случае и медицинских документов помогает в принятии более точных решений.
4. Телекоммуникации: В телекоммуникационной отрасли аналитические инструменты могут использоваться для анализа данных о поведении клиентов, определения тенденций использования услуг связи, выявления аномалий и прогнозирования спроса на новые услуги. Автоматическая обработка текстов из обзоров услуг и социальных медиа позволяет компаниям понять мнение клиентов и улучшить качество обслуживания.
5. Автомобильная промышленность: В автомобильной промышленности аналитика может использоваться для анализа данных о техническом обслуживании, диагностики неисправностей, прогнозирования спроса на автомобили и оптимизации производства. Анализ текстовых данных из отчетов о неисправностях, обзоров автомобилей и социальных медиа

позволяет производителям автомобилей улучшить качество и надежность своих продуктов.

Расширенная аналитика предоставляет широкий спектр возможностей для обработки и анализа текстовых данных в различных отраслях, что помогает предприятиям принимать обоснованные решения и достигать более эффективных результатов.

1.6 Задачи и техники NLP

Задачи и методы NLP часто используются для создания программного обеспечения на основе NLP. Это помогает разделить человеческий язык на понятные машине куски, которые можно использовать для создания программного обеспечения на основе NLP [22].

Ниже перечислены задачи, выполняемые в процессе NLP:

1. Stemming: Стэмминг – это процесс сокращения похожих слов до их основного слова. Основа слова может быть как значимой, так и бессмысленной.

Например:

- Finally, Final, Finalized - Final (meaningful)
- History, Historical - Histor (meaningless)
- Automates, Automatic, Automation - Automat (meaningless)
- Wait, Waiting, Waits - Wait (meaningful)

В результате сокращения похожих слов до одного стэма, обрабатываются вариации слова, такие как различные временные формы, числа или род. Это может быть полезно при анализе текстов, поиске информации или сокращении размера словарей в компьютерных приложениях.

В приведенных примерах, слова "Finally", "Final" и "Finalized" имеют общую основу "Final", которая сохраняет смысловую значимость. Слова "History" и "Historical" имеют базовую форму "Histor", которая не несет смысловой нагрузки. То же самое происходит с словами "Automates", "Automatic" и "Automation", которые имеют общую основу "Automat" без конкретного значения. Слова "Wait", "Waiting" и "Waits" имеют базовую форму "Wait", которая сохраняет смысл слова.

Стэмминг полезен для обработки текстовых данных, устранения разнообразия форм слов и сокращения словарного запаса, что может улучшить производительность и эффективность алгоритмов обработки естественного языка.

2. Lemmatization. Лемматизация – это процесс преобразования слов в их осмысленную базовую структуру. Основное различие между стеммингом и лемматизацией заключается в следующем:

Лемматизация преобразует слово в его значимую осмысленную базовую структуру, в то время как стемминг просто удаляет последнюю пару символов, которые могут быть или не быть значимыми.

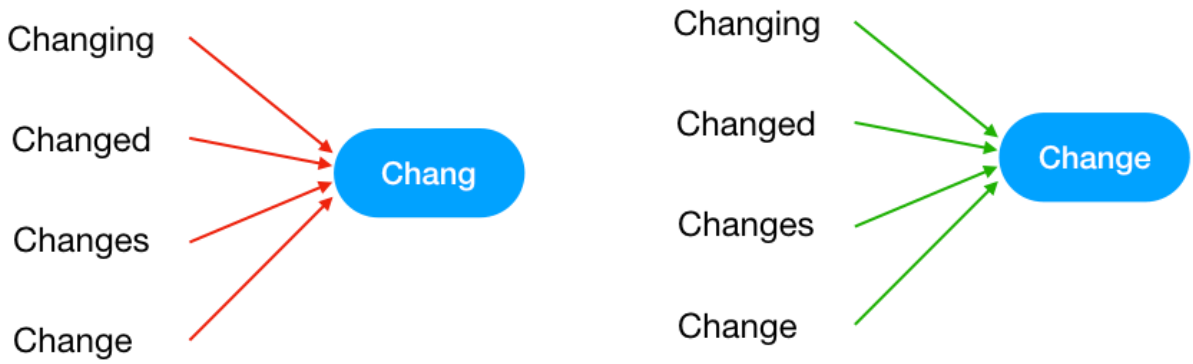


Рисунок 1. Stemming VS Lemmatization

Лемматизация является более сложным процессом, чем стемминг, поскольку она учитывает контекст и семантику слова (Рисунок 1) [23]. Цель лемматизации - привести слово к его базовой форме, называемой леммой, которая имеет смысловую значимость. Лемматизация учитывает грамматические правила языка, временные формы, род и число слова, а также его синтаксическую связь с другими словами в предложении.

Лемматизация позволяет извлечь более точную информацию из текстов и улучшить качество обработки естественного языка в различных задачах, таких как информационный поиск, машинный перевод, анализ тональности и многие другие. Благодаря сохранению семантической связи между словами, лемматизация обеспечивает более точное понимание текста и помогает избежать проблем, связанных с различными формами слов. Например, при лемматизации предложения "I am running late" слово "running" будет преобразовано в лемму "run", сохраняя его смысловую связь с глаголом "am". Это позволяет лучше понять действие, описанное в предложении, и обработать его соответствующим образом.

В целом, лемматизация предоставляет более полезную и осмысленную информацию о словах, чем стемминг, и является предпочтительным методом при работе с текстовыми данными во многих языковых задачах.

3. *Tokenization*. Токенизация – это процесс разбиения полного предложения на слова, и мы можем сказать, что каждое слово является токеном. Например, такое предложение, как «Hope you are doing well», имеет 5 токенов, т.е. Hope-you-are-doing-well. Точно так же токены могут быть как символами, так и подсловами.

Токенизация является важным шагом в обработке естественного языка, где полные тексты или предложения разбиваются на отдельные единицы, называемые токенами. Токенами могут быть слова, символы, подслова или другие элементы, в зависимости от контекста и используемого подхода.

В примере предложения "Hope you are doing well", токенизация разделяет его на пять токенов: "Hope", "you", "are", "doing" и "well". Каждый из этих токенов представляет собой отдельное слово. Токенизация помогает разделить текст на более мелкие части для дальнейшей обработки, анализа или применения в алгоритмах обработки естественного языка.

В некоторых случаях токены могут быть символами или подсловами [24]. Например, при использовании символьной токенизации предложение "Hope you are doing well" может быть разделено на отдельные символы: "H", "o", "p", "e", и так далее. При использовании техники подсловной токенизации, предложение может быть разделено на подслова, такие как "Hope", "you", "are", "do", "ing", "well", где некоторые из них являются фрагментами слов.

Токенизация играет важную роль в обработке текстовых данных, поскольку разделение текста на токены помогает в решении различных задач, таких как машинный перевод, распознавание речи, анализ тональности, классификация текста и другие. Токенизация также может быть основой для последующих шагов обработки, таких как лемматизация, удаление стоп-слов и извлечение признаков из текстовых данных.

4. *Удаление стоп-слов*: Стоп-слова [25] – это наиболее часто используемые слова в человеческом языке. Примерами стоп-слов в английском языке являются «a», «the», «is», «are» и т. д. Стоп-слова не содержат ценной информации, или мы можем сказать, что они не добавляют смысла предложению, поэтому стоп-слова могут быть игнорируются, не затрагивая смысла предложения. Точно так же в каждом человеческом языке есть стоп-слова.

5. Значение слова: Используется для определения того, могут ли одни и те же слова иметь разное значение в разных предложениях. Например:

The **bull** is running fast.

According to Senex, the **bull** is running high.

You should read this **book**, it has a great story

You should **book** the flight as soon as possible

Одни и те же слова «book» и «bull» использовались в приведенных выше примерах в разных контекстах. Таким образом, устранение неоднозначности смысла слова используется для понимания контекста [26], в котором слово использовалось в предложении.

Устранение неоднозначности смысла слова является важным аспектом в обработке естественного языка (NLP). Одни и те же слова могут иметь разное значение в разных контекстах, что может привести к неправильному пониманию текста. Для понимания контекста и определения правильного значения слова используются различные методы и подходы.

Один из подходов к устранению неоднозначности состоит в анализе окружающих слов и фраз в предложении. Смысл слова может быть определен на основе его связи с другими словами в контексте. Например, в предложении "The bull is running fast", слово "bull" имеет значение животного. В то же время, в предложении "According to Senex, the bull is running high", слово "bull" имеет

значение финансового рынка. Анализируя контекстные слова и фразы, можно определить правильное значение слова и понять его смысл.

Еще одним подходом к устранению неоднозначности является использование лексических баз данных или словарей семантических значений. Эти базы данных содержат информацию о значениях слов и их связях с другими словами. Например, словарь WordNet содержит семантические определения и связи между словами в английском языке. При обработке текста, система NLP может обращаться к таким словарям для определения правильного значения слова в контексте.

Также машинное обучение и статистические методы могут быть использованы для устранения неоднозначности смысла слова. Модели машинного обучения могут быть обучены на больших корпусах текстов, чтобы научиться предсказывать наиболее вероятное значение слова в данном контексте. Нейронные сети и алгоритмы глубокого обучения показывают хорошие результаты в решении этой задачи.

Будущее устранения неоднозначности смысла слова в NLP будет связано с развитием более точных и эффективных моделей и алгоритмов. С увеличением объема и разнообразия текстовых данных, доступных для обучения, модели смогут лучше улавливать контекст и предсказывать правильное значение слова. Также развитие методов обработки естественного языка, таких как генеративные модели и модели с учителем, может привести к дальнейшему улучшению в устранении неоднозначности смысла слова.

6. Тегирование части речи. Тегирование частью речи (Part Of Speech (POS) tagging) хорошо известно в NLP, оно используется для обозначения каждого слова в предложении или может быть абзацем с соответствующей частью речи. К частям речи относятся глаголы, наречия, прилагательные, местоимения и др.

Тегирование частей речи является одной из важных задач в области обработки естественного языка (NLP). Его целью является присвоение каждому слову в предложении или тексте соответствующей части речи, такой как глагол, существительное, прилагательное, наречие, местоимение и т.д. Тегирование частей речи имеет ключевое значение для понимания смысла текста и обработки естественного языка.

Одной из основных проблем в тегировании частей речи является омонимия и полисемия, то есть возможность одного слова иметь несколько разных частей речи в различных контекстах. Например, слово "банк" может быть существительным (место, где хранятся деньги) или глаголом (склоняться). Решение этой проблемы требует учета контекста и использования лингвистических правил или статистических моделей для принятия правильного решения о теге.

Для тегирования частей речи используются различные подходы. Одним из них является правила на основе лингвистических грамматик. Этот подход основан на определенных правилах, которые определяют, какие части речи могут сочетаться с определенными словами или группами слов. Например, глаголы могут сочетаться с существительными, а прилагательные могут модифицировать существительные.

С развитием глубокого обучения появились и новые подходы к тегированию частей речи. Нейронные сети, включая рекуррентные нейронные сети (RNN) и сверточные нейронные сети (CNN), показали высокую эффективность в задачах тегирования частей речи. Эти модели способны учитывать контекст предложения и использовать информацию о последовательности слов для определения наиболее вероятных тегов.

В будущем развитие тегирования частей речи будет направлено на повышение точности и эффективности моделей. Большая акцент будет сделан на применение глубокого обучения и использование больших корпусов текстов для обучения более точных моделей тегирования. Также возможно развитие более сложных алгоритмов, которые будут учитывать контекст не только в пределах предложения, но и в более широком контексте, учитывая предыдущие и последующие предложения.

2 Исследования в области обработки естественного языка

В области обработки естественного языка (Natural Language Processing, NLP) проводится множество научных исследований. Исследования оказали значительное влияние на различные области, включая машинный перевод, анализ текста, голосовые помощники, чат-боты и многое другое. Эта область активно развивается, и ниже приведены некоторые ключевые моменты и исследования, которые поставили основу для развития NLP.

Развитие моделей машинного обучения: С появлением глубоких нейронных сетей и методов глубокого обучения, NLP получила новые возможности. Модели, такие как рекуррентные нейронные сети (RNN), сверточные нейронные сети (CNN) и трансформеры, стали широко использоваться для задач обработки естественного языка, позволяя получать лучшие результаты в задачах, таких как машинный перевод, распознавание речи и анализ тональности текста. С появлением глубоких нейронных сетей и методов глубокого обучения, обработка естественного языка (NLP) пережила значительное развитие. Глубокие модели, такие как рекуррентные нейронные сети (RNN), сверточные нейронные сети (CNN) и трансформеры, стали широко применяться в NLP, и они привнесли значительные улучшения во множестве задач [27, 29].

RNN, благодаря своей способности улавливать последовательностные зависимости, стали основой для моделей, работающих с последовательностями текста. Они позволяют эффективно моделировать контекст и обрабатывать переменной длины входные данные. RNN успешно применяются в задачах, таких как машинный перевод, генерация текста и анализ тональности.

CNN, которые изначально были разработаны для обработки изображений, были адаптированы для работы с текстом. Они способны эффективно извлекать локальные и глобальные признаки из текстовых данных и использоваться для классификации текста, определения тональности и распознавания именованных сущностей.

Обработка последовательностей: Модели, способные обрабатывать последовательности, такие как рекуррентные нейронные сети (RNN) и трансформеры, стали основой для многих NLP-задач. Они способны улавливать контекстную информацию и обрабатывать переменной длины последовательности, что позволяет моделям более точно предсказывать и генерировать текст.

Обработка последовательностей является ключевым аспектом обработки естественного языка (NLP) и играет важную роль в решении различных задач, таких как машинный перевод, генерация текста, распознавание речи и анализ тональности текста. Два основных подхода к обработке последовательностей, которые привнесли существенный вклад в NLP, включают рекуррентные нейронные сети (RNN) и трансформеры.

Рекуррентные нейронные сети (RNN) являются одним из первых и наиболее распространенных подходов к обработке последовательностей в NLP. Они обладают способностью улавливать контекст и зависимости между элементами последовательности. RNN работают путем передачи информации от предыдущих состояний в следующие, что позволяет модели учитывать контекст и долгосрочные зависимости. Однако у RNN есть проблема затухающего и взрывающегося градиента, что может затруднить обучение модели на длинных последовательностях. Для решения этой проблемы были предложены модификации RNN, такие как Long Short-Term Memory (LSTM) и Gated Recurrent Unit (GRU), которые позволяют эффективно управлять потоком информации и избегать затухания/взрыва градиента [27].

2.1 Научные исследования в области NLP

Научная статья "Attention Is All You Need" (Васвани и др., 2017) [27] представляет собой ключевую работу в области NLP, которая внесла значительный вклад в развитие моделей машинного перевода. Она демонстрирует эффективность применения механизма внимания для обработки естественного языка и получает высокие результаты в задачах перевода и других NLP-задачах.

В этой статье авторы представляют модель Transformer, которая полностью основана на механизме внимания (attention). Ранее модели машинного перевода часто использовали рекуррентные нейронные сети (RNN), но модель Transformer предлагает новый подход, полностью отказываясь от RNN.

Модель Transformer основана на двух ключевых идеях: механизме внимания и концепции самовнимания (self-attention). Механизм внимания позволяет модели обращать внимание на разные части входного предложения при генерации соответствующего выхода. Самовнимание позволяет модели эффективно улавливать зависимости между словами в предложении, не зависимо от их относительного расположения.

Одним из ключевых преимуществ модели Transformer является ее способность работать параллельно и обрабатывать большие объемы текста более эффективно. Это делает ее особенно полезной для задач машинного перевода, где требуется обработка и генерация перевода для длинных предложений.

Статья "Attention Is All You Need" стала важным вехой в исследованиях NLP и имеет широкое применение не только в задачах машинного перевода, но и в других NLP-задачах, таких как распознавание речи, суммаризация текста и вопросно-ответные системы.

"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" (Devlin et al., 2018) [28]: Эта работа описывает модель BERT (Bidirectional Encoder Representations from Transformers), которая использует

маскированное языковое моделирование для предварительного обучения на больших объемах текста. BERT достигает выдающихся результатов в широком спектре задач NLP, таких как классификация текста, извлечение информации и вопросно-ответные системы. BERT имеет значительный вклад в области NLP, благодаря своей способности эффективно моделировать контекст и улавливать семантические отношения в тексте. Одним из ключевых преимуществ BERT является его способность понимать смысл слова в контексте, что помогает в обработке сложных задач, таких как классификация текста, извлечение информации и вопросно-ответные системы.

Предварительное обучение модели BERT на больших объемах данных позволяет ей захватывать богатые лингвистические характеристики и создавать универсальные представления для разных языковых задач. После предварительного обучения BERT может быть дообучен на относительно небольших размеченных наборах данных, что упрощает применение модели на практике.

Научная работа "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" имеет значительное значение для разработки NLP-систем, обеспечивая высокую точность и гибкость в различных задачах обработки естественного языка.

"Neural Machine Translation by Jointly Learning to Align and Translate" (Bahdanau et al., 2014) [29]: В этой работе представлена модель Seq2Seq с механизмом внимания, которая демонстрирует значительное улучшение в задаче машинного перевода. Модель Seq2Seq с механизмом внимания стала основой для многих последующих исследований в области машинного перевода и других задач NLP.

Модель Seq2Seq, которая стала основой данного исследования, состоит из двух основных компонентов: энкодера и декодера. Энкодер принимает на вход исходное предложение на одном языке и преобразует его в вектор фиксированной длины, который содержит сжатую семантическую информацию о предложении. Затем декодер принимает этот вектор и генерирует переведенное предложение на целевом языке.

Однако основным вкладом этой работы является предложенный механизм внимания. Вместо того, чтобы преобразовывать всю информацию в один вектор фиксированной длины, модель Seq2Seq с механизмом внимания позволяет ей "сосредоточиваться" на разных частях исходного предложения при генерации перевода. Это достигается путем вычисления весового коэффициента для каждого слова в исходном предложении, чтобы указать, насколько важно это слово для текущего генерируемого слова в переводе.

Механизм внимания позволяет модели лучше справляться с длинными предложениями и улучшает качество перевода. Он также открывает возможность для более гибких и интерпретируемых моделей, позволяя анализировать, на какие части исходного предложения модель обращает большее внимание при генерации перевода.

Вот еще несколько примеров научных работ в этой области:

1. "GloVe: Global Vectors for Word Representation" (Pennington et al., 2014) [30]: В этой работе представлен метод GloVe, который используется для получения векторных представлений слов. GloVe позволяет представлять слова в виде векторов в многомерном пространстве, учитывая статистические свойства их совместной встречаемости. Это векторное представление слов часто применяется в задачах NLP, таких как кластеризация и семантический анализ.
2. "Word2Vec" (Mikolov et al., 2013) [31]: В этой работе представлен метод Word2Vec, который позволяет получать векторные представления слов на основе их контекстного окружения. Word2Vec популяризировал подход к обучению эмбеддингов слов, который применяется в широком спектре NLP-задач, включая классификацию, кластеризацию и семантический анализ текста.

Это лишь несколько примеров исследовательских работ в области обработки естественного языка. Но эти работы являются важными вкладами в развитие NLP и существенно влияют на разработку новых методов и подходов для работы с текстовыми данными.

2.2 PolyAnalyst

PolyAnalyst [32] – это инструментарий для визуальной разработки и анализа данных и текстов, который обладает мощными возможностями работы с неструктурированными текстовыми данными. Он предоставляет возможность проводить анализ текстов, не требуя от пользователей навыков программирования.

С помощью PolyAnalyst пользователи могут создавать интерактивные отчеты и проводить различные аналитические задачи, связанные с текстовыми данными. Платформа предлагает гибкую визуальную среду разработки, что позволяет пользователям легко создавать сценарии анализа данных и текстов без необходимости программирования. PolyAnalyst предоставляет аналитикам и исследователям мощный инструментарий для работы с текстовыми данными, упрощая процесс анализа, классификации и извлечения информации из больших объемов текста на различных языках.

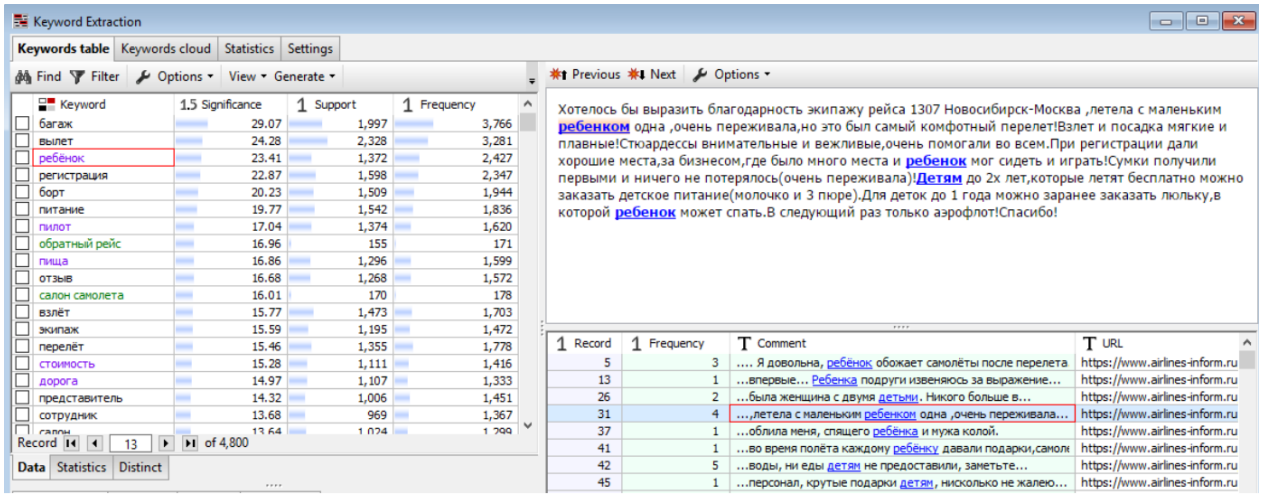


Рисунок 2. Поиск ключевых слов

Платформа PolyAnalyst (Рисунок 2) предлагает широкий набор инструментов для работы с текстом, включая классификацию и кластеризацию текстов, извлечение сущностей и фактов, определение тематики и тональности. Она способна обрабатывать документы на различных языках, включая русский и английский.

В процессе подготовки данных в платформе PolyAnalyst выполняются следующие этапы:

1. Извлечение информации: Платформа осуществляет извлечение ключевых слов, сущностей, фактов, а также анализ тональности текста. Это позволяет выделить важные элементы из текстовых данных и определить их связи и значения.
2. Классификация: PolyAnalyst предлагает возможность построения таксономий, то есть иерархических структур классификации. Также платформа позволяет проводить текстовую классификацию и кластеризацию текстовых данных. Это помогает организовать и категоризировать информацию в соответствии с заданными параметрами.
3. Поиск взаимосвязей: PolyAnalyst позволяет искать связи между терминами и терминологическими сетями. Также платформа предоставляет возможность мониторинга упоминаемости определенных терминов или фраз в текстовых данных.
4. Подготовка данных: В процессе подготовки данных PolyAnalyst выполняет несколько задач, включая определение языка текста, индексирование для ускорения поиска, проверку орфографии, удаление дубликатов и поиск общих фрагментов текста. Эти шаги помогают обеспечить точность и качество анализируемых данных.

PolyAnalyst является одним из ведущих инструментов анализа данных и автоматической обработки естественного языка (NLP) в современной индустрии. Он разрабатывается и поддерживается компанией Megaputer Intelligence, которая активно внедряет новые технологии и методы в своем продукте [33]. Развитие PolyAnalyst в области NLP было впечатляющим, приводя к значительному улучшению в его функциональности и способностях.

В начале своего развития PolyAnalyst предлагал базовые возможности анализа текста, такие как токенизация, удаление стоп-слов, выделение ключевых слов и определение частей речи. Однако с течением времени он стал сильно преуспевать и теперь предлагает более сложные функции, такие как выявление именованных сущностей, разрешение синонимов, извлечение отношений и классификацию текстов. Одним из основных достижений PolyAnalyst в области NLP является его способность анализировать тексты на естественном языке и извлекать из них ценную информацию. Например, он может автоматически распознавать сентимент (тональность) текста и классифицировать его как положительный, отрицательный или нейтральный. Это особенно полезно для компаний, которые хотят понять общественное мнение о своих продуктах или услугах. Еще одной важной функцией PolyAnalyst в области NLP является его способность работать с неструктурированными данными, такими как текстовые документы, электронные письма, отчеты и социальные медиа. Он может автоматически категоризировать и группировать документы, а также проводить анализ тематики для выявления наиболее релевантных тем и ключевых слов в них. Кроме того, PolyAnalyst предлагает возможности анализа текста на многих языках, что делает его универсальным инструментом для компаний с глобальным присутствием. Он поддерживает множество языковых моделей и может обрабатывать тексты на английском, испанском, французском, немецком и других популярных языках.

Следует отметить, что PolyAnalyst активно использует машинное обучение и глубокое обучение для улучшения своей производительности и точности анализа. Команда разработчиков постоянно обновляет и улучшает алгоритмы и модели, чтобы обеспечить лучший результат для пользователей.

В целом, развитие PolyAnalyst в области NLP является непрерывным процессом, и он продолжает эволюционировать, чтобы удовлетворить все более сложные требования бизнеса и исследований. Благодаря своим современным возможностям анализа текста и обработки естественного языка, PolyAnalyst продолжает оставаться одним из самых мощных и инновационных инструментов в этой области.

2.3 Elasticsearch

Elasticsearch [34] – это программное обеспечение с открытым исходным кодом, которое предоставляет возможности для поиска, сбора, анализа и хранения текстовых данных с использованием интеллектуальных алгоритмов. Оно предоставляет эффективные инструменты для работы с большими объемами данных (Big Data) и поставляется с модулем для создания кластеров, что позволяет проводить параллельный поиск и балансировку нагрузки на нескольких серверах.

Платформа Elasticsearch предлагает ряд компонентов, включающих [35]:

- Logstash: инструмент сбора, преобразования и сохранения событий из различных источников (файлы, базы данных, логи и т.д.) в реальном времени. Logstash обеспечивает надежный поток данных в Elasticsearch.
- Kibana: веб-интерфейс для взаимодействия с данными, хранящимися в Elasticsearch. Kibana предоставляет динамические панели мониторинга, таблицы, графики и диаграммы, которые отображают изменения в реальном времени и позволяют пользователю визуализировать и анализировать данные.
- FileBeat: агент, установленный на серверах, который собирает и отправляет различные типы оперативных данных в Elasticsearch. FileBeat позволяет передавать данные в режиме реального времени для дальнейшего анализа и хранения.

Elasticsearch автоматически создает индексы [36] и типы данных, что аналогично базе данных и таблицам в реляционных базах данных. Каждый тип данных имеет свою схему, известную как mapping, которая определяет структуру и типы данных полей. Elasticsearch не различает между одиночными значениями и массивами значений, и каждое поле может содержать как простое значение, так и массив строк.

Помимо Elasticsearch и PolyAnalyst, существует множество других аналогичных программ и инструментов, предназначенных для анализа и обработки текстовых данных в области обработки естественного языка (NLP). Вот несколько примеров таких программ:

1. Apache Lucene: Мощный поисковый движок, на котором базируется Elasticsearch. Lucene предоставляет возможности индексации и поиска текстовых данных, а также поддерживает множество функций NLP, включая токенизацию, стемминг и извлечение сущностей.
2. NLTK (Natural Language Toolkit): Библиотека для языка программирования Python, предназначенная для работы с текстовыми данными и выполнения различных задач NLP. NLTK предоставляет широкий спектр функций, включая токенизацию, лемматизацию, тегирование частей речи, анализ синтаксиса и многое другое.
3. Stanford CoreNLP: Платформа обработки естественного языка, разработанная в Университете Стэнфорда. CoreNLP предлагает различные модули для анализа текста, включая сегментацию предложений, токенизацию, лемматизацию, анализ синтаксиса, выделение именованных сущностей и другие функции.
4. OpenNLP: Библиотека с открытым исходным кодом, предназначенная для разработки программ на основе NLP. OpenNLP предлагает набор инструментов для различных задач NLP, включая токенизацию, сегментацию предложений, определение частей речи, извлечение именованных сущностей и многое другое.
5. GATE (General Architecture for Text Engineering): Инструментарий и разработочная платформа для обработки и анализа текстовых данных. GATE предоставляет множество ресурсов и инструментов для NLP, включая токенизацию, морфологический анализ, анализ семантики, извлечение информации и другие функции.

Каждая из этих программ имеет свои особенности, и выбор наиболее подходящего инструмента зависит от конкретных требований и задач, которые необходимо решить в области NLP. Эти программы обеспечивают разработчиков и исследователей возможность эффективно обрабатывать и анализировать текстовые данные, открывая широкие возможности в области обработки естественного языка.

2.4 Alem Media Monitoring

Alem Media Monitoring (Алем ММ) [37] является программным решением, разработанным для анализа социального мнения в интернет-пространстве. Ее основная цель - помочь компаниям и организациям получить важные инсайты о том, как их продукты, услуги или репутация воспринимаются пользовательским сообществом в онлайн-среде.

Программа Alem ММ предоставляет мощные инструменты и технологии для мониторинга и анализа информации, размещенной в различных источниках, включая социальные сети, блоги, форумы и новостные сайты. Она использует передовые методы обработки естественного языка (NLP) и машинного обучения для извлечения и классификации тональности, эмоций и тенденций в отношении определенной компании, бренда или темы.

Алем ММ предлагает множество полезных функций, таких как:

- Мониторинг социальных медиа: Программа сканирует и анализирует сообщения и публикации, размещенные в социальных сетях, идентифицируя тональность (позитивную, негативную, нейтральную) и эмоциональные оттенки высказываний.
- Анализ трендов: Alem ММ помогает определить популярные темы, обсуждаемые в онлайн-среде, и выявляет ключевые слова и фразы, связанные с определенной компанией или продуктом.
- Идентификация влиятельных лиц: Программа помогает определить пользователей с большим количеством подписчиков и влиянием в социальных сетях, что позволяет компаниям сфокусироваться на ключевых мнениях и влиятельных лидерах мнений.
- Создание отчетов: Alem ММ предоставляет возможность генерировать подробные отчеты с визуализацией данных, диаграммами и графиками, которые помогают визуализировать и анализировать результаты мониторинга социального мнения.

Программа Alem ММ может быть полезна компаниям в различных отраслях, таких как маркетинг, реклама, общественные связи и исследования рынка, помогая им принимать более информированные решения и реагировать на изменения в социальном мнении.

2.5 Потенциал развития NLP

С развитием искусственного интеллекта (ИИ) и глубокого обучения, NLP становится все более совершенным и обладает огромным потенциалом для революции в различных областях, включая коммуникацию, бизнес, здравоохранение, автоматизацию и многое другое. Вот некоторые из того, до чего может дойти развитие NLP в будущем.

Продвижение в понимании естественного языка: Одной из ключевых целей в развитии NLP является создание систем, способных полноценно понимать и интерпретировать естественный язык, так же, как это делает человек. В будущем ожидается развитие глубоких моделей, которые будут способны уловить семантическую и прагматическую стороны языка, учитывать контекст и даже обладать способностью к эмоциональному восприятию.

Усовершенствование автоматического перевода: Машинный перевод является одной из важных областей NLP, и его развитие будет продолжаться. Ожидается, что в будущем модели машинного перевода будут все более точными и способными переводить с высокой степенью лингвистической и культурной точности между различными языками. Благодаря использованию глубокого обучения и нейронных сетей, автоматический перевод станет более естественным и практичным для повседневного использования.

Развитие контекстуальных моделей: Одной из сложностей в NLP является правильное понимание и интерпретация контекста. В будущем ожидается развитие моделей, которые способны учитывать более широкий контекст, включая предыдущие предложения, диалоги или документы в целом. Это позволит системам NLP делать более информированные выводы и предоставлять более точные ответы на запросы пользователей.

Расширение области применения NLP: В будущем NLP будет использоваться во множестве новых областей и приложений. Например, в сфере здравоохранения, системы NLP могут анализировать и обрабатывать медицинские документы, истории болезни и симптомы пациентов для предоставления точных диагнозов и рекомендаций по лечению. В бизнесе, NLP может применяться для автоматического анализа рынка, анализа настроений клиентов, создания персонализированных рекомендаций и многое другое.

Учет культурных и лингвистических особенностей: Каждый язык и культура имеют свои уникальные особенности и нюансы. В будущем NLP будет все более адаптироваться к различным языкам и культурам, учитывая их специфические грамматические правила, идиомы, метафоры и культурные контексты. Это позволит создавать более точные и культурно-чувствительные системы NLP для межкультурного общения и взаимодействия.

Интерактивность и диалоговые системы: В будущем ожидается развитие более интерактивных и диалоговых систем NLP. Это позволит пользователям взаимодействовать с компьютерами и устройствами естественным образом, задавать вопросы, получать объяснения и решения проблем в режиме реального

времени. Такие системы могут иметь огромное применение в образовании, технической поддержке, путешествиях и других сферах жизни.

Этические вопросы и нормативные аспекты: В связи с широким применением NLP возникают и этические вопросы, связанные с приватностью данных, биасами в алгоритмах, авторскими правами и другими аспектами. В будущем важно разрабатывать нормативные и этические стандарты для использования NLP, чтобы обеспечить безопасность, справедливость и учет интересов всех сторон.

3 Практическая часть

В рамках разработки системы NLP будут включены несколько ключевых этапов, необходимых для достижения поставленных целей.

Для хранения и обработки данных, полученных из социальных сетей и новостных порталов, будет создана система хранения парсированных данных и обработанных аналитических результатов. Это позволит сохранять информацию для последующего анализа и использования.

Одной из ключевых задач системы будет определение тональности текста комментариев с новостных порталов и социальных сетей. Для этого будет разработан словарь настроений на русском и казахском языках, который позволит оценивать тональность комментариев по анализируемым темам. Это позволит системе определить, является ли комментарий положительным, отрицательным или нейтральным.

Одним из важных шагов в разработке системы будет создание аналитического приложения с удобной количественной и качественной графической визуализацией результатов мониторинга. Такая визуализация позволит пользователям системы легко интерпретировать и понимать полученные результаты анализа текста комментариев.

Для разработки системы будет использоваться Jupyter - популярная среда для интерактивного программирования, которая позволяет проводить исследования и разработку на языке программирования Python. Python предоставляет богатый набор инструментов и библиотек для работы с текстовыми данными и обработки естественного языка.

В целом, разработка системы NLP для определения тональности текста комментариев с новостных порталов и социальных сетей требует комплексного подхода и включает в себя использование API, разработку системы хранения данных, создание словарей настроений, аналитическое приложение и удобный интерфейс. Применение современных технологий и инструментов позволит достичь поставленных целей и обеспечить эффективную обработку и анализ текстовых данных.

3.1 Датасет и его обзор

Система использует словари тональности русского и казахского языков и алгоритмы машинного обучения для определения тональности текстов социальных сетей. Также представлена вся структура и функциональные возможности системы. Для построения моделей машинного обучения используется датасеты с содержанием тем и комментариев на русском и

казахском языках. Затем производительность моделей оценивается с помощью показателей точности, полноты и гармонической меры (precision, recall, f1-score).

```
]: Topics_all
```

]:

	Unnamed: 0.1	Unnamed: 0	id	text	id.1	lan	name	cleaned_text
0	1	1	3353	столица, построенная елбасы	2	ru	Положительная	столиц построен елбас
1	3	3	3358	назарбаев и "две коровы" - неудобная...	2	ru	Положительная	назарба и quot две коров quot неудобн правд дл...
2	4	4	3350	назарбаев коке не обижай мою маму! меньше тира...	2	ru	Положительная	назарба кок не обижа мо мам меньш тиран твор с...
3	5	5	3359	купи две коровы. речь елбасы	2	ru	Положительная	куп две коров реч елбас
4	8	8	3362	елбасы поставил точку в разговорах о досрочных...	2	ru	Положительная	елбас постав точк в разговор о досрочн выбор в...
...	...	...	...	...	...	...	...	...
1495	5432	5432	28169	с начала 2019 года в казахстане совершено 799 ...	1	ru	Отрицательная	с нача год в казахстан соверш факт изнасилован...
1496	5433	5433	49804	"власть не делит виновных на казахов и дунган"...	1	ru	Отрицательная	власт не дел виновн на казах и дунга глав миор...
1497	5435	5435	38249	позор правительства, дороги лёд, ужас ! митинг	1	ru	Отрицательная	позор правительств дорог л д ужас митинг
1498	5437	5437	38250	по данным оон, в казахстане от бытового насили...	1	ru	Отрицательная	по дан оон в казахстан от бытов насил ежегодн ...
1499	5438	5438	38251	власти ирака создали "кризисные ячейки", чтобы...	1	ru	Отрицательная	власт ирак созда кризисн ячейк чтоб попыта вос...

3000 rows × 8 columns

Рисунок 3. Датасет с темами на русском языке

Ввод [12]:
```
Kaz_topics
```

Out[12]:

	Unnamed: 0	Unnamed: 0.1	Unnamed: 0.1.1	Unnamed: 0.1.1.1	Unnamed: 0.1.1.1.1	id	text	tonal_id	language_id	quant	cleaned_text
0	0	0	0	0	1	922	статистикалық мәліметтерге сүйенсек еліміздегі...	0	2	1	статистик мәлімет сүйен е ір кәсіпо компания о...
1	1	1	1	1	3	1033	астанада жыл басынан адам ашық құдыққа түсіп к...	0	2	1	ас жыл ба а ашық құ түс кет ело әкімдік tengri...
2	2	2	2	2	4	1307	в нью йорке двух казахстанцев обвиняют в участ...	0	2	1	в нью йор двух казахстанцев обвиняют в участии...
3	3	3	3	3	5	1310	в министерстве иностранных дел проверяют прина...	0	2	1	в министерств иностранных дел проверяют принад...
4	4	4	4	4	7	1671	департамент агентства рк по делам государствен...	0	2	1	департамент агентств рк по де государственно с...
...	...	...	...	...	...	...	...	...	...	...	...
462	462	462	462	462	1198	5573	осыдан жыл қазақстан ұлттық валютасы құнсыздан...	1	2	1	о жыл қазақс ұлт валю құн ерк айнал жіберіл е ...
463	463	463	463	463	1199	5578	желтоқсан және қаңтар айларында еліміздің басы...	1	2	1	желтоқсан жән қаң ай ел бас ау ө солтус өң қат...
464	464	464	464	464	1200	5585	егер бұған базалық зейнетақы ең төменгі күнкөр...	1	2	1	е бұ баз зейн ең те күнкөріс дең тең төлен ең ...
465	465	465	465	465	1201	5586	автокөліксуйер қауым айлыпұлды басына түскен б...	1	2	4	автокөліксу қ айлыпұл ба тус бір бәлекет кер р...
466	466	466	466	466	1202	5590	informburo kz үй жануарларын ұшақта немесе пой...	1	2	6	informburo kz үй жану ұ не пойыз ал жүру қан ш...

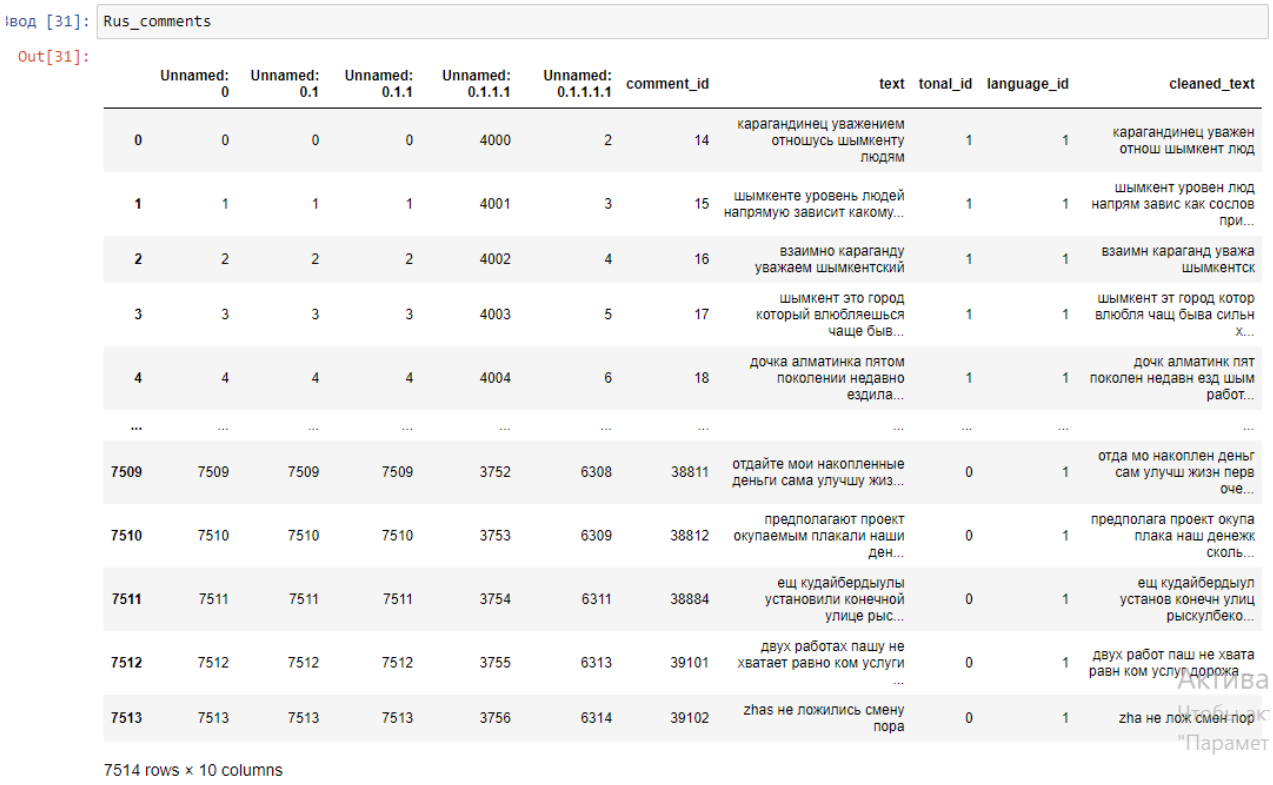
Рисунок 4. Датасет с темами на казахском языке

Для анализа тональности тем (Рисунок 3,4) и комментариев, извлеченных из различных источников, таких как новостные порталы (например, Казпресс, Тенгри Ньюс) и социальные сети (например, ВКонтакте, Instagram, Facebook), используется методика обработки текста и машинного обучения.

Собранные комментарии могут содержать разные мнения и эмоциональные оттенки, включая положительные, негативные и нейтральные.

Анализ тональности помогает определить эмоциональную окраску каждого комментария и классифицировать их соответственно.

Для этого в процессе анализа текста применяются методы обработки естественного языка (Natural Language Processing, NLP). Можно использовать алгоритмы машинного обучения, такие как анализ сентимента на основе нейронных сетей, методы классификации. Анализ тональности комментариев позволяет понять общее настроение пользователей относительно конкретной темы или события, провести анализ общественного мнения и оценить реакцию на новости или события. Это может быть полезным для мониторинга общественного мнения, репутации бренда, анализа рынка или принятия стратегических решений.



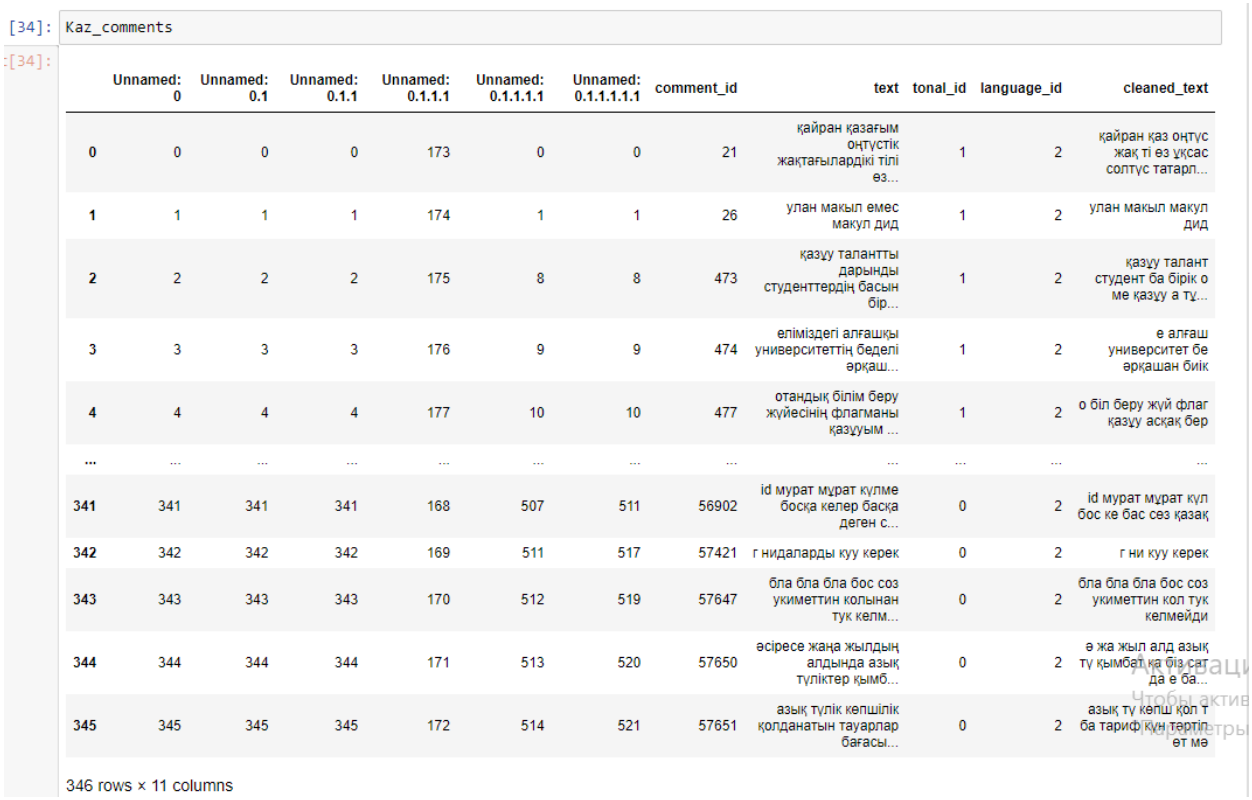
Ввод [31]:
```
Rus_comments
```

Out[31]:

	Unnamed: 0	Unnamed: 0.1	Unnamed: 0.1.1	Unnamed: 0.1.1.1	Unnamed: 0.1.1.1.1	comment_id	text	tonal_id	language_id	cleaned_text
0	0	0	0	4000	2	14	карагандинец уважением отношусь шымкенту людям	1	1	карагандинец уважен отношу шымкент люд
1	1	1	1	4001	3	15	шымкенте уровень людей напрямую зависит какому...	1	1	шымкент уровен люд напрям завис как сослов при...
2	2	2	2	4002	4	16	взаимно караганду уважаем шымкентский	1	1	взаимн караганд уважа шымкентск
3	3	3	3	4003	5	17	шымкент это город который влюбляешься чаще быв...	1	1	шымкент эт город котор влюбля чащ быва сильн х...
4	4	4	4	4004	6	18	дочка алматинка пятом поколении недавно ездила...	1	1	дочк алматинк пят поколен недавн езд шым работ...
...	...	...	...	...	...	...	...	...	...	...
7509	7509	7509	7509	3752	6308	38811	отдайте мои накопленные деньги сама улучшу жиз...	0	1	отда мо накоплен деньг сам улучш жизн перв оче...
7510	7510	7510	7510	3753	6309	38812	предполагают проект окупаемым плакали наши ден...	0	1	предполага проект окупа плака наш денежк сколь...
7511	7511	7511	7511	3754	6311	38884	ещ кудайбердыулы установили конечной улице рыс...	0	1	ещ кудайбердыул установ конечн улиц рыскулбеко...
7512	7512	7512	7512	3755	6313	39101	двух работах пашу не хватает равно ком услуги ...	0	1	двух работ паш не хвата равн ком услуг дорожа
7513	7513	7513	7513	3756	6314	39102	zhas не ложились смену пора	0	1	zha не лож смен пор

7514 rows × 10 columns

Рисунок 5. Датасет с комментариями на русском языке



[34]: Kaz_comments

[34]:

	Unnamed: 0	Unnamed: 0.1	Unnamed: 0.1.1	Unnamed: 0.1.1.1	Unnamed: 0.1.1.1.1	Unnamed: 0.1.1.1.1.1	comment_id	text	tonal_id	language_id	cleaned_text
0	0	0	0	173	0	0	21	қайран қазағым оңтүстік жақтағылардікі тілі өз...	1	2	қайран қаз оңтүс жақ ті өз ұқсас солтүс татарл...
1	1	1	1	174	1	1	26	улан мақыл емес макул дид	1	2	улан макыл макул дид
2	2	2	2	175	8	8	473	қазұу талантты дарынды студенттердің басын бір...	1	2	қазұу талант студент ба бірік о ме қазұу а тұ...
3	3	3	3	176	9	9	474	еліміздегі алғашқы университеттің беделі әрқаш...	1	2	е алғаш университет бе әрқашан бик
4	4	4	4	177	10	10	477	отандық білім беру жүйесінің флагманы қазұуым ...	1	2	о біл беру жүй флаг қазұу асқақ бер
...	...	...	...	...	...	...	...	...	...	...	...
341	341	341	341	168	507	511	56902	id мурат мұрат күлме босқа келер басқа деген с...	0	2	id мурат мұрат күл бос ке бас сез қазақ
342	342	342	342	169	511	517	57421	г нидаларды куу керек	0	2	г ни куу керек
343	343	343	343	170	512	519	57647	бла бла бла бос соз укиметтин колынан тук келм...	0	2	бла бла бла бос соз укиметтин кол тук келмейди
344	344	344	344	171	513	520	57650	әсіресе жаңа жылдың алдында азық түліктер қымб...	0	2	ә жа жыл алд азық ту қымбат қа біз сат да е ба...
345	345	345	345	172	514	521	57651	азық түлік көпшілік қолданатын тауарлар бағасы...	0	2	азық ту көпш қол т ба тариф күн тәртіп өт мә

346 rows × 11 columns

Рисунок 6. Датасет с комментариями на казахском языке

Анализ позволит выявить общественные социальные настроения в городах Алматы и Нур-Султан и других крупных областных городах Казахстана.

Тональность комментариев (Рисунок 5,6) под новостными порталами в Казахстане может быть разнообразной и зависит от множества факторов, таких как тема новости, политическая ситуация, настроения общества и т.д. Однако, можно выделить несколько основных тенденций в тональности комментариев, которые часто встречаются.

Положительная тональность: Некоторые комментарии могут быть положительно настроены и выражать поддержку или одобрение. Это может быть связано с успехами страны, положительными изменениями в обществе, достижениями отдельных личностей и другими благоприятными событиями. В таких комментариях можно найти слова благодарности, похвалы, восхищения и эмоциональную поддержку.

Негативная тональность: Некоторые комментарии могут быть критическими и негативно настроенными. Это может быть связано с проблемами в обществе, неудовлетворенностью действиями властей или отдельных личностей, экономическими проблемами и другими негативными событиями. В таких комментариях можно найти выражение недовольства, разочарования, обвинения и требования изменений.

Политическая тональность: Политическая обстановка в Казахстане имеет большое влияние на тональность комментариев. Комментарии могут быть приверженными той или иной политической партии или лидеру, выражать поддержку или критику политических решений, а также содержать

политические дебаты и споры. Такие комментарии могут быть страстными, эмоциональными и иногда агрессивными.

Нейтральная тональность: Некоторые комментарии могут быть нейтральными и содержать объективные факты или просто выражать мнение без ярко выраженных эмоций. В таких комментариях можно найти вопросы, просьбы о дополнительной информации, обмен мнениями и т.д.

Важно отметить, что тональность комментариев под новостными порталами в Казахстане может меняться в зависимости от конкретной новости, времени публикации, активности пользователей и других факторов. Однако, в целом, комментарии могут отражать разнообразные точки зрения и эмоциональные реакции на события, происходящие в стране.

3.2 Архитектура

Функциональность спроектированной системы реализована в компонентах:

1. Источники данных (Data sources): представлены новостными порталами, блогами и социальными сетями.
2. Модуль коннектора (Connector): используется для подключения к источникам данных.
3. Модуль лингвистического конструктора (linguistic constructor): используется для создания словарей настроений, включающих слова, принадлежащие к любому из трех классов: положительные, отрицательные и нейтральные.
4. Модуль анализа и обработки данных (Data analysis and processing): использует словари настроений и алгоритмы машинного обучения для SA. Кроме того, этот модуль создает социальную аналитику, определяющую социальные настроения.
5. Модуль результатов (Results): содержит сформированную реляционную базу текстов и комментариев, аналитические отчеты, графики и таблицы.

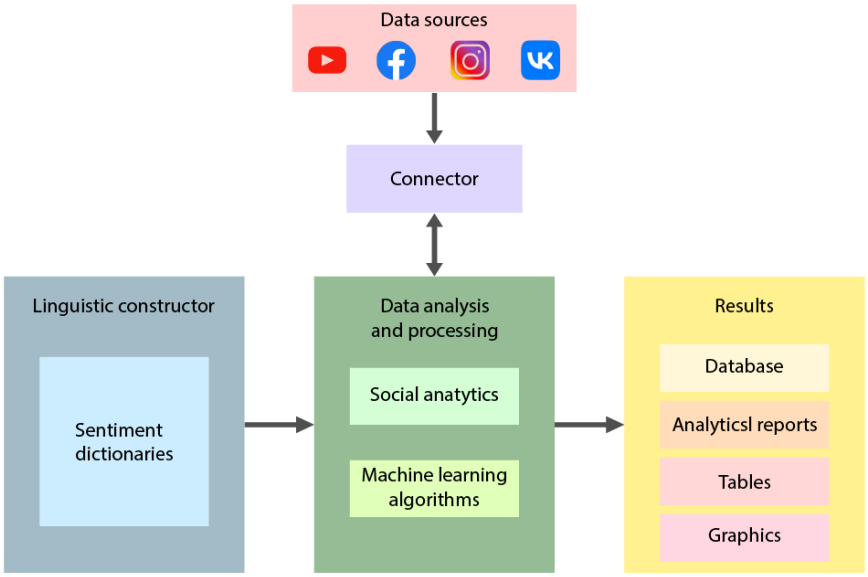


Рисунок 7. Архитектура

Лингвистический модуль (Рисунок 7) определяет тональность текстов с использованием сформированных словарей тональностей. Здесь используется функция, вычисляющая тональность по максимальному количеству положительных, отрицательных и нейтральных слов в тексте. Эффективность этого подхода во многом зависит от качества разработанного словаря настроений. Хотя этот подход очень эффективен, создание высококачественного словаря тональности требует больших усилий. После того, как исходный словарь тональности создан вручную, он расширяется за счет синонимов и антонимов из более крупных словарей, существующих для многих языков.

В системе будет разработаны большие словари тональностей для русского и казахского языков. Следующая формула находит тональность текста:

$$S_t=\langle Max(w_{pos},w_{neut},w_{neg}),D\rangle, \quad (1.1)$$

где S_t — тональность текста; w_{pos} - количество положительных слов; w_{neut} - количество нейтральных слов; w_{neg} - количество отрицательных слов; D - словарь настроений.

Инструмент анализа тональности, маркирующий тексты и комментарии пользователей по трем классам настроений (положительные, нейтральные и отрицательные), будет являться основной частью. Классы тональности назначаются с использованием гибридного подхода: на основе лексики (словари тональности) и на основе машинного обучения. Подход на основе лексики присваивает метку по наибольшему количеству слов одного из трех классов тональности. Подход, основанный на машинном обучении, использует обученные модели машинного обучения с наивысшей эффективностью с точки

зрения точности, полноты и гармонической меры (precision, recall, f1-score), такие как NB, LR, SVM, k-NN, Decision tree (DT), Random Forest (RF) и XGBoost.

$$S_t = \langle M, T \rangle, \quad (1.2)$$

где S_t — тональность текста; M — ML-модель; T — текстовый документ.

```
Ввод [15]: stop_words_russian = set(stopwords.words('russian'))
           stop_words_kazakh = set(stopwords.words('kazakh'))

Ввод [16]: words_to_remove = []
           for i in stop_words_russian:
               if i=='не' or i=='ни':
                   words_to_remove.append(i)

           words_to_remove2 = []
           for i in stop_words_kazakh:
               if i=='жоқ' or i=='емес':
                   words_to_remove2.append(i)

Ввод [17]: words_to_remove
Out[17]: ['не', 'ни']

Ввод [18]: words_to_remove2
Out[18]: ['емес', 'жоқ']
```

Рисунок 8. Стоп-слова

Происходит создание двух списков (Рисунок 8): words_to_remove и words_to_remove2. Он используется для удаления определенных слов из других списков stop_words_russian и stop_words_kazakh.

Первый цикл начинается со списка stop_words_russian. Для каждого слова i в этом списке проверяется, является ли оно "не" или "ни". Если слово равно "не" или "ни", то оно добавляется в список words_to_remove.

Аналогично, второй цикл начинается со списка stop_words_kazakh. Для каждого слова i в этом списке проверяется, является ли оно "жоқ" или "емес". Если слово равно "жоқ" или "емес", то оно добавляется в список words_to_remove2.

Многие алгоритмы Machine Learning очень чувствительны к шумам. В NLP к таким шумам относятся стоп-слова. Чаще всего это слова, которые не несут большой информативной нагрузки. Чтобы освободить место в базе данных и сократить время обработки текста, из него удаляются неинформативные слова, которые не представляют ценности для NLP.

3.3 Data processing

Разделение датасета на положительные (Рисунок 9) и негативные (Рисунок 10) примеры является одним из ключевых шагов в анализе текста и машинном обучении. Этот процесс выполняется с целью создания обучающего набора данных для разработки моделей классификации или сентимент-анализа. В задачах сентимент-анализа, например, датасеты могут содержать отзывы пользователей или тексты социальных медиа сообщений. Разделение на положительные и негативные примеры позволяет обучить модель классифицировать тексты как выражающие позитивное или негативное отношение.

Ввод [3]:
```
Rus_topics_pos = Rus_topics[Rus_topics['tonal_id'] == 1]
```

Ввод [4]:
```
Rus_topics_pos
```

Out[4]:

	Unnamed: 0	Unnamed: 0.1	Unnamed: 0.1.1	Unnamed: 0.1.1.1	Unnamed: 0.1.1.1.1	id	text	tonal_id	language_id	quant	cleaned_text
0	0	0	0	520	0	17	казахстане города республиканского значения ас...	1	1	2	казахстан город республиканск значен астан алм...
1	1	1	1	521	1	18	компромисс обеспечением надлежащего контроля ш...	1	1	4	компромисс обеспечен надлежа контрол школьн ст...
2	2	2	2	522	4	22	астане подвели итоги конкурса творческая семья...	1	1	1	астан подвел итог конкурс творческ сем гран за...
3	3	3	3	523	9	30	православные мира отмечают крещение господне а...	1	1	1	православн мир отмеча крещен господн астан про...
4	4	4	4	524	11	32	нурлан сейтимов назначен ответственным секрета...	1	1	1	нурла сейтим назнач ответствен секретар внешне...
...	...	...	...	...	...	...	...	...	...	...	...
515	515	515	515	1035	823	2484	руководителем департамента государственных дох...	1	1	1	руководител департамент государствен доход шым...
516	516	516	516	1036	824	2488	летний аким туркисбского района алматы павел к...	1	1	1	летн ак туркисбск район алмат павел кулагин же...
517	517	517	517	1037	827	2503	менять шины автолюбителям казахстана рассказал...	1	1	1	меня шин автолюбител казахста рассказа комитет...
518	518	518	518	1038	830	2514	астане наблюдаются безветренная погода обильны...	1	1	1	астан наблюда безветрен погод обильн снегопад ...
519	519	519	519	1039	834	2527	нұр отан усилит контроль реализацией послания ...	1	1	1	нур ота усил контрол реализац послан президент...

520 rows × 11 columns

Рисунок 9. Rus_topic_pos

```
Ввод [5]: Rus_topics_neg = Rus_topics[Rus_topics['tonal_id'] == 0]
Ввод [6]: Rus_topics_neg
Out[6]:
```

	Unnamed: 0	Unnamed: 0.1	Unnamed: 0.1.1	Unnamed: 0.1.1.1	Unnamed: 0.1.1.1.1	id	text	tonal_id	language_id	quant	cleaned_text
520	520	520	520	0	2	19	актобе кондуктор накричала бабушку выгнала авт...	0	1	1	актоб кондуктор накрича бабушк выгна автобус в...
521	521	521	521	1	3	21	преподаватель физкультуры который избил студен...	0	1	1	преподавател физкультур котор изб студент колл...
522	522	522	522	2	5	26	тоо алматыэнергосбыт не смогло договориться уп...	0	1	1	то алматыэнергосб не смогл договор управля ком...
523	523	523	523	3	6	27	петропавловске летняя девочка получила компрес...	0	1	1	петропавловск летн девочк получ компрессион пе...
524	524	524	524	4	7	28	казахстане разводится каждая третья пара слова...	0	1	1	казахстан развод кажд трет пар слов специалист...
...	...	...	...	...	...	...	...	...	...	...	...
1035	1035	1035	1035	515	1199	5663	американский индекс nasdaq обвалился процента ...	0	1	9	американск индекс nasdaq обвал процент явля кр...
1036	1036	1036	1036	516	1201	5666	курс нацвалюты составил тенге доллар итогам ут...	0	1	4	курс нацвалют состав тенг доллар итог утрен то...

Рисунок 10. Rus_topics_neg

Это важно для множества приложений, включая анализ отзывов, определение общественного мнения, выявление эмоциональной тональности и т.д. Создание обучающего набора с разметкой позволяет модели учиться на примерах и находить общие признаки, связанные с положительными или негативными контекстами. Разделение датасета также позволяет оценить качество модели на тестовых данных, что важно для понимания ее производительности и возможности применения в реальных сценариях. Комбинирование положительных и негативных примеров обеспечивает балансировку и позволяет создать репрезентативный обучающий набор для моделирования и анализа текста.

Перед началом обучения модели, было проведено разделение датасета на две тональности (где ['tonal_id'] == 0 является негативной тональностью, а ['tonal_id'] == 1 -положительной).

```
Ввод [7]: Rus_topics_pos = Rus_topics_pos[:520]
Ввод [8]: Topics_rus = pd.concat([Rus_topics_pos, Rus_topics_neg])
Ввод [9]: Topics_rus
Out[9]:
```

	Unnamed: 0	Unnamed: 0.1	Unnamed: 0.1.1	Unnamed: 0.1.1.1	Unnamed: 0.1.1.1.1	id	text	tonal_id	language_id	quant	cleaned_text
0	0	0	0	520	0	17	казахстане города республиканского значения ас...	1	1	2	казахстан город республиканск значен астан алм...
1	1	1	1	521	1	18	компромисс обеспечением надлежащего контроля ш...	1	1	4	компромисс обеспечен надлежа контрол школьн ст...
2	2	2	2	522	4	22	астане подвели итоги конкурса творческая семья...	1	1	1	астан подвел итог конкурс творческ сем гран за...
3	3	3	3	523	9	30	православные мира отмечают крещение господне а...	1	1	1	православн мир отмеча крещен господн астан про...
4	4	4	4	524	11	32	нурпан сейтимов назначен ответственным секрета...	1	1	1	нурла сейтим назнач ответствен секретар внешне...
...	...	...	...	...	...	...	...	...	...	...	...
1035	1035	1035	1035	515	1199	5663	американский индекс nasdaq обвалился процента ...	0	1	9	американск индекс nasdaq обвал процент явля кр...
1036	1036	1036	1036	516	1201	5666	курс нацвалюты составил тенге доллар итогам ут...	0	1	4	курс нацвалют состав тенг доллар итог утрен то...
1037	1037	1037	1037	517	1202	5667	курс нацвалюты составил тенге доллар итогам ут...	0	1	4	курс нацвалют состав тенг доллар итог утрен то...
1038	1038	1038	1038	518	1205	5670	павлодаре весенняя распутица превратила некото...	0	1	4	павлодар весен распутиц преврат некотор улиц г...
1039	1039	1039	1039	519	1208	5673	крахе доллара тенге окажется тяжелой ситуации ...	0	1	1	крах доллар тенг окажет тяжел ситуац так мнен ...

1040 rows × 11 columns

Рисунок 11. Объединение Rus_topics_pos и Rus_topics_neg

Здесь происходит выбор первых 520 строк из DataFrame Rus_topics_pos и сохранение их в переменную Rus_topics_pos (Рисунок 11). Это предполагает, что DataFrame Rus_topics_pos содержит более 520 строк, и операция выбирает только первые 520 строки для дальнейшей обработки.

Далее используется функция concat() из библиотеки pandas для объединения двух DataFrame: Rus_topics_pos и Rus_topics_neg. Результат объединения сохраняется в переменную Topics_rus. Rus_topics_pos и Rus_topics_neg содержат одинаковые структуры данных (столбцы), и объединение выполняется по вертикали, то есть строки из Rus_topics_pos и Rus_topics_neg будут скомбинированы в одном DataFrame Topics_rus.

В случае задачи классификации, где требуется обучить модель на данных с различными классами (положительные и негативные тональности), важно иметь сбалансированный набор данных. Это означает, что количество примеров каждого класса должно быть примерно одинаковым или близким к этому.

Если разница в количестве примеров между классами слишком велика, это может привести к смещению модели в сторону более представленного класса. Модель может стать предвзятой и иметь проблемы с обобщением на меньше представленный класс.

Поэтому, для достижения баланса в обучающем наборе данных, можно использовать различные подходы, такие как увеличение или уменьшение количества примеров в каждом классе. В данном случае, код Rus_topics_pos = Rus_topics_pos[:520] выбирает первые 520 положительных примеров, чтобы сделать их количество сравнимым с количеством негативных примеров, затем объединяет их в Topics_rus.

Обеспечение равного количества примеров в каждом классе позволяет модели получить более сбалансированное представление данных и более объективно обучиться на обоих классах. Это может привести к лучшей производительности модели при классификации новых, ранее не виденных примеров.

Тот же процесс выполняется для датасета с заголовками на казахском языке.

Объединение положительных и негативных комментариев в один выполняется с целью создания универсального и сбалансированного набора данных для анализа тональности и других задач обработки текста. Это позволяет сбалансированно представить оба класса комментариев, что важно для обучения моделей с высокой производительностью и точностью.

На языке Python это можно сделать, например, с помощью библиотеки pandas, которая предоставляет удобные инструменты для работы с данными. Можно загрузить положительные и негативные комментарии в две отдельные структуры данных (например, списки или датафреймы) и затем объединить их в один датасет, добавив метку класса для каждого комментария (например, 1 для положительных и 0 для негативных комментариев).

```
Ввод [21]: Topics_kaz = pd.concat([Kaz_topics_pos, Kaz_topics_neg])

Ввод [22]: Kaz_topics
```

Out[22]:

	Unnamed: 0	Unnamed: 0.1	Unnamed: 0.1.1	Unnamed: 0.1.1.1	Unnamed: 0.1.1.1.1	id	text	tonal_id	language_id	quant	cleaned_text
0	0	0	0	0	1	922	статистикалық мәліметтерге сүйенсек еліміздегі...	0	2	1	статистик мәлімет сүйен е ір кәсіпо компания о...
1	1	1	1	1	3	1033	астанада жыл басынан адам ашық құдыққа түсіп к...	0	2	1	ас жыл ба а ашық құ түс кет ело әкімдік tengri...
2	2	2	2	2	4	1307	в нью йорке двух казахстанцев обвиняют в участ...	0	2	1	в нью йор двух казахстанцев обвиняют в участии...
3	3	3	3	3	5	1310	в министерстве иностранных дел проверяют прина...	0	2	1	в министерств иностранных дел проверяют принад...
4	4	4	4	4	7	1671	департамент агентства рк по делам государствен...	0	2	1	департамент агентств рк по де государственно с...
...	...	...	...	...	...	...	...	...	...	...	...
462	462	462	462	462	1198	5573	осыдан жыл қазақстан ұлттық валютасы құнсыздан...	1	2	1	о жыл қазақс ұлт валю құн ерк айнал жіберіп е ...
463	463	463	463	463	1199	5578	желтоқсан және қаңтар айларында еліміздің басы...	1	2	1	желтоқсан жән қаң ай ел бас ау ә солтүс өң қат...
464	464	464	464	464	1200	5585	егер бұған базалық зейнетақы ең төменгі күнкөр...	1	2	1	е бұ баз зейн ең тө күнкөріс дең тең төлен ең ...
465	465	465	465	465	1201	5586	автокөліксүйер қауым айыппұлды басына түскен б...	1	2	4	автокөліксу қ айыппұл ба түс бір бөлекет көр р...
466	466	466	466	466	1202	5590	informburo kz үй жануарларын ұшақта немесе пой...	1	2	6	informburo kz үй жану ұ не пойыз ал жүру қан ш...

467 rows × 11 columns

Рисунок 12. Объединение Kaz_topics_pos и Kaz_topics_neg

Объединение положительных и негативных комментариев в один датасет (Рисунок 12) не только облегчает дальнейший анализ данных, но и позволяет обучать модели на сбалансированной выборке, что может привести к более точным и надежным результатам при анализе тональности и классификации комментариев.

```
Ввод [30]: Rus_comments = pd.concat([Rus_comments_pos, Rus_comments_neg])

Ввод [31]: Rus_comments
```

Out[31]:

	Unnamed: 0	Unnamed: 0.1	Unnamed: 0.1.1	Unnamed: 0.1.1.1	Unnamed: 0.1.1.1.1	comment_id	text	tonal_id	language_id	cleaned_text
0	0	0	0	4000	2	14	карагандинец уважением отношусь шымкенту людям	1	1	карагандинец уважен отнош шымкент люд
1	1	1	1	4001	3	15	шымкенте уровень людей напрямую зависит какому...	1	1	шымкент уровен люд напрям завис как сослов при...
2	2	2	2	4002	4	16	взаимно караганду уважаем шымкентский	1	1	взаимн караганд уважа шымкентск
3	3	3	3	4003	5	17	шымкент это город который влюбляешься чаще быв...	1	1	шымкент эт город котор влюбля чащ быва сильн х...
4	4	4	4	4004	6	18	дочка алматинка пятом поколении недавно ездила...	1	1	дочк алматинк пят поколен недавн езд шым работ...
...	...	...	...	...	...	...	...	...	...	...
7509	7509	7509	7509	3752	6308	38811	отдайте мои накопленные деньги сама улучшу жиз...	0	1	отда мо накоплен деньг сам улучш жизн перв оче...
7510	7510	7510	7510	3753	6309	38812	предполагают проект окупаемым плакали наши ден...	0	1	предполага проект окупа плака наш денежк сколь...
7511	7511	7511	7511	3754	6311	38884	ещ кудайбердыулы установили конечной улице рыс...	0	1	ещ кудайбердыул установ конечн улиц рыскулбеко...
7512	7512	7512	7512	3755	6313	39101	двух работах пашу не хватает равно ком услуги ...	0	1	двух работ паш не хвата равн ком услуг дорожа ...
7513	7513	7513	7513	3756	6314	39102	zhas не ложились смену пора	0	1	zha не лож смен пор

7514 rows × 10 columns

Рисунок 13. Объединение Rus_comments_pos и Rus_comments_neg

Объединение положительных и негативных комментариев на русском языке (Рисунок 13).

```
Ввод [41]: Comments_kaz = pd.concat([Kaz_comments_pos, Kaz_comments_neg])

Ввод [42]: Comments_kaz
```

Out[42]:

	Unnamed: 0	Unnamed: 0.1	Unnamed: 0.1.1	Unnamed: 0.1.1.1	Unnamed: 0.1.1.1.1	Unnamed: 0.1.1.1.1.1	comment_id	text	tonal_id	language_id	cleaned_text
0	0	0	0	173	0	0	21	қайран қазағым оңтүстік жақтағылардікі тілі өз...	1	2	қайран қаз оңтус жақ ті өз ұқсас солтүс татарл...
1	1	1	1	174	1	1	26	улан мақыл емес мақул дид	1	2	улан мақыл мақул дид
2	2	2	2	175	8	8	473	қазұу талантты дарынды студенттердің басын бір...	1	2	қазұу талант студент ба бірік о ме қазұу а тұ...
3	3	3	3	176	9	9	474	еліміздегі алғашқы университеттің беделі өрқаш...	1	2	е алғаш университет бе өрқашан биік
4	4	4	4	177	10	10	477	отандық білім беру жүйесінің флагманы қазұуым ...	1	2	о біл беру жүй флаг қазұу асқақ бер
...	...	...	...	...	...	...	...	...	...	...	...
341	341	341	341	168	507	511	56902	id мурат мұрат күлме босқа келер басқа деген с...	0	2	id мурат мұрат күл бос ке бас сез қазақ
342	342	342	342	169	511	517	57421	г нидаларды куу керек	0	2	г ни куу керек
343	343	343	343	170	512	519	57647	бла бла бла бос соз укиметтин қолынан тук келм...	0	2	бла бла бла бос соз укиметтин кол тук келмейди
344	344	344	344	171	513	520	57650	әсіресе жаңа жылдың алдында азық түліктер қымб...	0	2	ә жа жыл алд азық ту қымбат қа біз сат да е ба...
345	345	345	345	172	514	521	57651	азық түлік көпшілік қолданатын тауарлар бағасы...	0	2	азық ту көпш қол т ба тариф күн тәртіп өт мә

щелкните чтобы развернуть вывод; двойной щелчок чтобы скрыть вывод

346 rows × 11 columns

Рисунок 14. Объединение Kaz_comments_pos и Kaz_comments_neg

Объединение положительных и негативных комментариев на казахском языке (Рисунок 14).

3.4 Обработка текстовых данных в датафрейме Topics_rus и Comment_rus

```
Ввод [24]: for index, row in Topics_rus.iterrows():
               row['text'] = re.sub('[^a-zA-Za-яА-ЯӘІҢҒҮҰҚӨәіңғүұқөһ]',' ',row['text'])
               tokens = word_tokenize(row['text'])
               tokens2 = [t for t in tokens if t not in stop_words_russian]
               Topics_rus.at[index, 'text'] = ' '.join(tokens2)

Ввод [25]: for index, row in Comments_rus.iterrows():
               row['text'] = re.sub('[^a-zA-Za-яА-ЯӘІҢҒҮҰҚӨәіңғүұқөһ]',' ',row['text'])
               tokens = word_tokenize(row['text'])
               tokens2 = [t for t in tokens if t not in stop_words_russian]
               Comments_rus.at[index, 'text'] = ' '.join(tokens2)
```

Рисунок 15. Обработка текстовых данных в датафрейме Topics_rus и Comment_rus

Цикл начинается с итерирования по строкам датафрейма (Рисунок 15). Для каждой строки выполняется следующее:

- re.sub('[^a-zA-Za-яА-ЯӘІҢҒҮҰҚӨәіңғүұқөh]',' ',row['text']) – здесь используется регулярное выражение для замены всех символов, которые не являются буквами латинского или кириллического алфавита, на пробелы. Это позволяет удалить все знаки препинания, цифры и другие символы, которые могут мешать обработке текста.
- tokens = word_tokenize(row['text']) - здесь используется библиотека nltk для токенизации текста, то есть разбиения текста на отдельные слова (токены). Результатом будет список токенов.
- tokens2 = [t for t in tokens if t not in stop_words_russian] – здесь происходит удаление стоп-слов из списка токенов tokens. Список стоп-слов для русского языка, stop_words_russian, был предварительно определен. Это можно сделать, например, с помощью библиотеки nltk или с использованием собственного списка стоп-слов.
- Topics_rus.at[index, 'text'] = ' '.join(tokens2) – здесь происходит замена текста в исходной строке датафрейма Topics_rus на обработанный текст, который получился после удаления символов, токенизации и удаления стоп-слов. Функция join() используется для объединения списка токенов tokens2 в одну строку, разделенную пробелами. Используется метод at[] для обновления значения ячейки в датафрейме, соответствующей текущей строке и столбцу 'text'.

```
In 32  1  x_topics_train_rus, x_topics_test_rus, y_topics_train_rus, y_topics_test_rus = train_test_split(Topics_rus['cleaned_text'],
           Topics_rus['tonal_id'], test_size=0.2, random_state=50)

In 33  1  x_topics_train_kaz, x_topics_test_kaz, y_topics_train_kaz, y_topics_test_kaz = train_test_split(Topics_kaz['cleaned_text'],
           Topics_kaz['tonal_id'], test_size=0.2, random_state=50)

In 34  1  x_comments_train_rus, x_comments_test_rus, y_comments_train_rus, y_comments_test_rus = train_test_split
           (Comments_rus['cleaned_text'], Comments_rus['tonal_id'], test_size=0.2, random_state=50)

In 35  1  x_comments_train_kaz, x_comments_test_kaz, y_comments_train_kaz, y_comments_test_kaz = train_test_split
           (Comments_kaz['cleaned_text'], Comments_kaz['tonal_id'], test_size=0.2, random_state=50)
```

Рисунок 16. Разделение данных на обучающую и тестовую выборки

Исходные данные находятся в датафрейме Topics_rus. Этот датафрейм содержит два столбца: cleaned_text, содержащий предобработанные тексты для анализа, и tonal_id, содержащий метки тональности для каждого текста.

Функция train_test_split() (Рисунок 16) из библиотеки sklearn используется для разделения данных на обучающую и тестовую выборки. Аргумент test_size указывает на размер тестовой выборки в процентах (20% в данном случае), а random_state задает начальное значение для генератора случайных чисел. Это позволяет получать одинаковый результат при каждом запуске кода.

Результатом выполнения этой строки кода являются четыре переменные: x_topics_train_rus содержит предобработанные тексты для обучения модели, y_topics_train_rus содержит метки тональности для обучения, x_topics_test_rus содержит предобработанные тексты для тестирования модели, а y_topics_test_rus содержит метки тональности для тестирования.

```
tokenizer = Tokenizer(num_words=20000)
tokenizer.fit_on_texts(x_topics_train_rus)

x_topics_train_rus = tokenizer.texts_to_sequences(x_topics_train_rus)
x_topics_test_rus = tokenizer.texts_to_sequences(x_topics_test_rus)
```

```
tokenizer2 = Tokenizer(num_words=20000)
tokenizer2.fit_on_texts(x_topics_train_kaz)

x_topics_train_kaz = tokenizer.texts_to_sequences(x_topics_train_kaz)
x_topics_test_kaz = tokenizer.texts_to_sequences(x_topics_test_kaz)
```

```
tokenizer3 = Tokenizer(num_words=20000)
tokenizer3.fit_on_texts(x_comments_train_rus)

x_comments_train_rus = tokenizer3.texts_to_sequences(x_comments_train_rus)
x_comments_test_rus = tokenizer3.texts_to_sequences(x_comments_test_rus)
```

```
tokenizer4 = Tokenizer(num_words=20000)
tokenizer4.fit_on_texts(x_comments_train_kaz)

x_comments_train_kaz = tokenizer4.texts_to_sequences(x_comments_train_kaz)
x_comments_test_kaz = tokenizer4.texts_to_sequences(x_comments_test_kaz)
```

Рисунок 17. Преобразование текстовых данных в числовой формат

Этот код используется для преобразования текстовых данных в числовой формат (Рисунок 17), который может быть использован моделями машинного обучения.

Tokenizer из библиотеки keras используется для токенизации текстовых данных, то есть разделения текста на отдельные слова или токены. Аргумент num_words=20000 указывает, что мы хотим ограничить количество уникальных слов (или токенов) до 20000 наиболее часто встречающихся в обучающих данных.

- fit_on_texts() метод используется для обучения токенизатора на текстовых данных обучающей выборки. Он создает словарь, где каждому слову (токену) присваивается уникальный целочисленный идентификатор.
- texts_to_sequences() метод используется для преобразования текстовых данных в последовательности целых чисел, используя ранее созданный словарь. В результате каждому слову (токену) в тексте сопоставляется его соответствующий идентификатор из словаря.

В данном случае, x_topics_train_rus и x_topics_test_rus содержат предобработанные текстовые данные, которые были разделены на обучающую и тестовую выборки, соответственно. После обучения токенизатора на обучающей выборке, texts_to_sequences() метод используется для преобразования каждой из этих выборок в последовательности целых чисел. Результатом являются два новых набора данных, x_topics_train_rus и x_topics_test_rus, которые содержат последовательности целых чисел, представляющие тексты в числовой форме.

```
embedding_matrix = zeros((vocab_size, 300))
```

```
embedding_matrix
```

```
array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]])
```

Рисунок 18. Векторное представление слов

В обработке естественного языка (NLP) матрица обычно используется для хранения векторных представлений слов (word embeddings) (Рисунок 18). Векторные представления слов – это плотные векторные представления слов, которые могут захватывать семантическую и синтаксическую информацию о словах. Каждая строка матрицы соответствует слову в словаре, а столбцы соответствуют размерности пространства векторов, в котором представлены векторные представления слов.

В данном случае матрица имеет 300 столбцов, что означает, что каждое слово в словаре будет представлено вектором размерности 300.

```
model = Sequential()

embedding_layer = Embedding(vocab_size, 300, weights=[embedding_matrix], input_length=maxlen)
model.add(embedding_layer)
model.add(Conv1D(128, 5, activation='relu'))
model.add(GlobalMaxPooling1D())
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))
```

Рисунок 19. Создание модели Sequential

Создается объект модели Sequential с помощью model = Sequential() (Рисунок 19).

Создается слой Embedding с размером входного словаря vocab_size, размерностью векторных представлений 300, весами embedding_matrix и максимальной длиной входной последовательности maxlen. Этот слой будет использоваться для преобразования входных последовательностей слов в векторные представления слов.

Добавляется слой Conv1D с 128 фильтрами, размером ядра 5, функцией активации 'relu'. Этот слой будет использоваться для извлечения признаков из векторных представлений слов.

Добавляется слой GlobalMaxPooling1D, который будет использоваться для выбора наиболее важных признаков из полученных векторов признаков.

Добавляется слой Dropout с коэффициентом 0.5, который будет использоваться для регуляризации модели.

Добавляется полносвязный слой Dense с 1 нейроном и функцией активации 'sigmoid'. Этот слой будет использоваться для классификации текстов на два класса (положительный и отрицательный).

```
from tensorflow.keras.metrics import Precision, Recall

precision = Precision()
recall = Recall()

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=[precision, recall])

history = model.fit(x_topics_train_rus, y_topics_train_rus, batch_size=64, epochs=10, verbose=1, validation_split=0.2)
```

```
Epoch 1/10
11/11 [==============================] - 13s 612ms/step - loss: 0.6937 - precision: 0.0000e+00 - recall: 0.0000e+00 - val_loss:
0.7048 - val_precision: 0.0000e+00 - val_recall: 0.0000e+00
Epoch 2/10
11/11 [==============================] - 6s 557ms/step - loss: 0.6898 - precision: 1.0000 - recall: 0.0157 - val_loss: 0.6957 -
val_precision: 0.0000e+00 - val_recall: 0.0000e+00
прокрутки вывода; двойной щелчок - чтобы скрыть
- 6s 561ms/step - loss: 0.6850 - precision: 1.0000 - recall: 0.0502 - val_loss: 0.6981 -
val_precision: 1.0000 - val_recall: 0.0208
Epoch 4/10
11/11 [==============================] - 6s 561ms/step - loss: 0.6710 - precision: 1.0000 - recall: 0.1317 - val_loss: 0.7014 -
val_precision: 1.0000 - val_recall: 0.0312
Epoch 5/10
11/11 [==============================] - 6s 587ms/step - loss: 0.6330 - precision: 0.6647 - recall: 0.3605 - val_loss: 0.7067 -
val_precision: 0.5779 - val_recall: 0.9271
Epoch 6/10
11/11 [==============================] - 7s 602ms/step - loss: 0.5965 - precision: 0.5831 - recall: 0.5392 - val_loss: 0.7447 -
val_precision: 0.5732 - val_recall: 0.9375
Epoch 7/10
```


Рисунок 20. Тестовые результаты обучения нейронной сети

Этот код используется для компиляции и обучения нейронной сети на задаче классификации текстов на два класса (положительные и отрицательные) (Рисунок 20).

Первые две строки импортируют метрики точности (Precision) и полноты (Recall) из модуля tensorflow.keras.metrics. Эти метрики будут использоваться для оценки качества модели в процессе обучения.

Далее определяется модель, которая состоит из встроенных слоев Keras, таких как Embedding, Conv1D, GlobalMaxPooling1D и Dense. Модель компилируется с помощью метода compile(), который принимает три аргумента:

loss='binary_crossentropy' - функция потерь, используемая для обучения модели. В данном случае используется бинарная кросс-энтропия, так как задача классификации двух классов.

optimizer='adam' - алгоритм оптимизации, используемый для обучения модели. В данном случае используется оптимизатор Adam, который является эффективным алгоритмом оптимизации стохастического градиентного спуска.

metrics=[precision, recall] - список метрик, используемых для оценки качества модели в процессе обучения. В данном случае используются метрики точности и полноты.

Затем модель обучается с помощью метода fit(), который принимает в качестве аргументов обучающие данные (x_topics_train_rus и y_topics_train_rus), размер пакета (batch_size), количество эпох (epochs), а также различные параметры, управляющие процессом обучения, такие как verbose (уровень вывода), validation_split (доля данных, используемых для проверки модели в каждой эпохе), и другие.

```
print("Russian topics classification score: ", score[0])
print("Russian topics classification accuracy: ", score[1])
print("Russian topics classification precision: ", score[2])
print("Russian topics classification recall: ", score[3])
print("Russian topics classification F-measure", 2*score[2]*score[3]/(score[2]+score[3]))
```

```
Russian topics classification score:  0.7170617580413818
Russian topics classification accuracy:  0.699999988079071
Russian topics classification precision:  0.13333334028720856
```

Рисунок 21. Результаты обучения

Обучение модели происходит на обучающих данных, а метрики precision и recall вычисляются на каждой эпохе как оценки качества модели (Рисунок 21). Результаты обучения сохраняются в переменной history.

Ввод [84]:
```
plt.title("ROC")
plt.plot(fpr_topics_rus, tpr_topics_rus, 'green', label = 'ROC for Russian texts = %0.2f' % roc_auc_topics_rus)
plt.legend(loc = 'lower right')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```

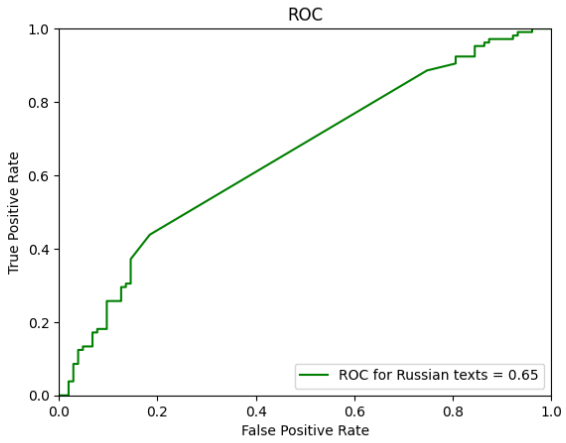


Рисунок 22. График модели классификации текстов на русском языке

Данный код строит ROC-кривую (Рисунок 22) и вычисляет площадь под кривой (AUC) для модели классификации текстов на русском языке . ROC-кривая показывает, какое количество верных положительных ответов (true positives) можно получить при различных уровнях ложных положительных ответов (false positives). AUC показывает, насколько хорошо модель классификации способна различать классы на основе вероятности, которую она присваивает каждому объекту. Чем выше значение AUC, тем лучше качество модели.

```
Ввод [164]: print("Kazakh topics classification score: ", score2[0])
            print("Kazakh topics classification accuracy: ", score2[1])
            print("Kazakh topics classification precision: ", score2[2])
            print("Kazakh topics classification recall: ", score2[3])
            print("Kazakh topics classification F-measure", 2*score2[2]*score2[3]/(score2[2]+score2[3]))

Kazakh topics classification score:  0.47470225045021547
Kazakh topics classification accuracy:  0.7978723442300837
Kazakh topics classification precision:  0.9999999944444444
Kazakh topics classification recall:  0.48648648517165816
Kazakh topics classification F-measure 0.6545454521652893
```

Рисунок 23. Результаты обучения на казахских темах

В представленных результатах оценки классификации по казахским темам есть несколько метрик, которые характеризуют производительность модели (Рисунок 23). Оценка (score) классификации по казахским темам составляет 0.4747. Это значение может варьироваться от 0 до 1, где ближе к 1 означает более высокую производительность модели. Точность (accuracy) классификации по казахским темам равна 0.7979. Эта метрика показывает, какая доля предсказаний модели является правильными.

Precision классификации по казахским темам составляет 0.9999. Она измеряет долю истинно положительных предсказаний модели относительно всех положительных предсказаний. Полнота (recall) классификации по казахским темам равна 0.4865. Эта метрика отражает долю истинно положительных предсказаний модели относительно всех действительно положительных примеров в данных. F-мера (F-measure) классификации по казахским темам составляет 0.6545. Она является гармоническим средним между точностью и полнотой и представляет собой обобщенную метрику для оценки производительности модели.

Исходя из этих результатов, можно сделать вывод, что модель показывает высокую точность (accuracy), но у нее низкая полнота (recall) и F-мера (F-measure). Это может указывать на то, что модель достаточно хорошо определяет отрицательные примеры, но упускает некоторое количество положительных примеров. Для улучшения результатов, возможно, потребуется дополнительная настройка модели или изменение алгоритма классификации.

```
6]: plt.title("ROC")
    plt.plot(fpr_topics_kaz, tpr_topics_kaz, 'red', label = 'ROC for Kazakh texts = %0.2f' % roc_auc_topics_kaz)
    plt.legend(loc = 'lower right')
    plt.xlim([0, 1])
    plt.ylim([0, 1])
    plt.ylabel('True Positive Rate')
    plt.xlabel('False Positive Rate')
    plt.show()
```

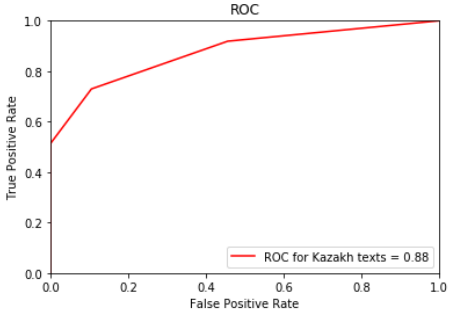


Рисунок 24. График модели классификации текстов на казахском языке

В данном случае, значение AUC равное 0.88, (Рисунок 24) указывает на хорошую производительность модели в классификации казахских текстов. Чем ближе AUC к 1, тем лучше модель разделяет классы. Это может означать, что модель имеет высокую способность правильно классифицировать положительные и отрицательные примеры казахских текстов.

График ROC визуализирует соотношение между чувствительностью (True Positive Rate) и специфичностью (False Positive Rate) модели при различных пороговых значениях. Чувствительность показывает, какая доля положительных примеров была правильно классифицирована, а специфичность показывает, какая доля отрицательных примеров была неправильно классифицирована. Таким образом, результат ROC для казахских текстов равный 0.88 свидетельствует о достаточно хорошей производительности модели и ее способности правильно разделять классы.

ЗАКЛЮЧЕНИЕ

Сравнительный анализ зарубежных аналитических платформ и систем Казахстана позволил сделать вывод, что зарубежные аналитические платформы по большей части нацелены на продвижение бизнеса и бренда. При этом они охватывают лишь информационное пространство зарубежных стран и мало ориентированы на существующие социальные проблемы. Существующие платформы аналитики iMAS, Alem Media Monitoring уделяют больше внимания анализу общественного мнения по широкому кругу политических и социально-экономических вопросов. Они стремятся охватить наиболее актуальные темы в больших и малых диапазонах времени и используют алгоритмы машинного обучения для быстрого и эффективного определения тональности текстов и комментариев пользователей.

Система будет отслеживать текущую политическую и социально-экономическую ситуацию в стране, позволяет искать ключевые слова по любым интересующим темам, определяет настроения тем с помощью подходов словаря и алгоритмов машинного обучения, а также определяет социальное самочувствие на основе таких показателей, как уровень активности обсуждения темы в обществе, уровень интереса к теме в обществе, уровень социального настроения.

Анализируя результаты обучения на данных на русском языке можно заметить, что метрика точности на обучающих данных оставалась на уровне 1.0, но на валидационных данных была низкой. Это может свидетельствовать о переобучении модели на обучающих данных. Также метрики полноты на валидационных данных были низкими, что означает, что модель не могла правильно определить положительные классы. Модель показала средний результат на задаче классификации текстов на темы на русском языке с F1-мерой 0.72 и точностью 0.7. Это может быть улучшено путем изменения параметров модели и/или препроцессинга данных.

ROC-кривая показывает, как изменяется отношение между верно-положительными и ложно-положительными решениями при изменении порога классификации. Высокое значение площади под ROC-кривой (ROC AUC) свидетельствует о хорошей производительности модели. Модель имеет ROC AUC в размере 0.76, что является хорошим показателем производительности.

Однако, низкое значение точности указывает на то, что модель часто дает ложно-положительные результаты, что может быть нежелательным в некоторых приложениях. Например, если модель используется для автоматической классификации спама в электронных письмах, ложно-положительные результаты могут привести к блокировке легитимных писем. Поэтому, возможно, стоит пересмотреть порог классификации или использовать другие методы оценки модели, такие как кривая точности-отзыва.

Исходя из результатов оценки классификации по казахским темам, можно сделать следующие заключения.

Модель показывает высокую точность (accuracy) в размере 0.7979, что означает, что примерно 79.79% предсказаний модели являются правильными. Это говорит о том, что модель обладает способностью правильно классифицировать большинство примеров. Однако, полнота (recall) модели составляет всего 0.4865, что означает, что модель способна обнаружить только примерно 48.65% действительно положительных примеров в данных. Это может свидетельствовать о том, что модель упускает значительную часть положительных примеров. F-мера (F-measure), которая является обобщенной метрикой для оценки производительности модели, составляет 0.6545. Это значение указывает на среднюю гармоническую пропорцию между точностью и полнотой. Низкое значение F-меры также указывает на недостаточную способность модели одновременно достигать высокой точности и полноты.

Таким образом, можно сделать вывод, что модель имеет высокую точность, но низкую полноту и F-меру. Это может свидетельствовать о том, что модель успешно определяет отрицательные примеры, но испытывает сложности в обнаружении и классификации положительных примеров.

Для улучшения результатов классификации возможно потребуется провести дополнительную настройку модели или рассмотреть возможность изменения алгоритма классификации. При анализе данных стоит обратить внимание на причины низкой полноты и F-меры, возможно, в данных присутствуют особенности или несбалансированность классов, которые затрудняют классификацию положительных примеров. Также, стоит рассмотреть использование других методов и техник машинного обучения, которые могут лучше справиться с задачей классификации по казахским темам.

В заключение, NLP продолжает развиваться и играть важную роль во многих областях, от обработки естественного языка до автоматического перевода и голосовых помощников. Однако, чтобы достичь оптимальных результатов в различных вариантах использования, требуется постоянное исследование и инновации. Глубокое обучение, синтаксическое и семантическое обучение являются ключевыми компонентами NLP, помогающими улучшить точность, полноту и качество классификации и понимания текстовых данных.

Кроме того, NLP продолжает слияние с другими областями, такими как компьютерное зрение и робототехника, что создает новые возможности для разработки более полных и универсальных систем. К примеру, сочетание NLP с компьютерным зрением может помочь создать системы, способные анализировать и понимать мультимодальные данные, такие как изображения и тексты, для решения сложных задач в различных областях.

В целом, будущее NLP обещает больше инноваций, улучшений и расширения возможностей. С постоянным развитием новых моделей, алгоритмов и подходов, а также с учетом потребностей и контекстов различных языков и культур, NLP продолжит играть важную роль в создании продуктов и услуг, которые эффективно работают с естественным языком и удовлетворяют потребности пользователей.

Список литературы

1 Эстебан ОО. Рост социальных сетей. Наш мир в данных, 2019.
2 Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. A neural probabilistic language model. Journal of machine learning research, 2003.
3 Collobert, R., & Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning, 2008.
4 Chowdhury, G. Natural language processing. Annual Review of Information Science and Technology, 2003.
5 Moore, R. C. Practical Natural Language Processing by Computer, 2018.
6 Обработка естественного языка. SAS, 2020.
7 Автоматическая обработка текстов на естественном языке и анализ данных: учеб. пособие / Большакова Е.И., Воронцов К.В., Ефремова Н.Э., Клышинский Э.С., Лукашевич Н.В., Сапин А.С. — М.: Изд-во НИУ ВШЭ, 2017. — 269 с.
8 Chomsky, N. Syntactic Structures. The Hague: Mouton, 1957.
9 Somers, H. Machine Translation: Latest Developments. In: The Oxford Handbook of Computational Linguistics. Mitkov R. (ed.). Oxford University Press, 2003, p. 512-528.
10 Brown P., Pietra S., Mercer R., Pietra V. The Mathematics of Statistical Machine Translation. // Computational Linguistics, Vol. 19(2): 263-3
11 Strzalkowski, T. (ed.) Natural Language Information Retrieval. Kluwer,19p.
12 Г., П. Методы автоматизированной обработки текстов. – М.: ИПИ РАН, 2008.
13 Harabagiu, S., Moldovan D. Question Answering. In: The Oxford Handbook of Computational Linguistics. Mitkov R. (ed.). Oxford University Press, 2003, p. 560-582.
14 Grishman R. Information extraction. In: The Oxford Handbook of Computational Linguistics. Mitkov R. (ed.). Oxford University Press, 2003, p. 545-559.
15 BARTON, E. et al. Computational Complexity and NLP. Cambridge: MIT Press, 1986.
16 Anna Espunya i Prat. Computational linguistics: a brief introduction. Dpt. de Filologia Anglesa i Gerrnanística Universitat Autbnoma de Barcelona
17 Кобозева И.М. Лингвистическая семантика. — М., 2009.
18 Касевич В.Б. Элементы общей лингвистики. — М.: Наука, 1977.
19 Лингвистический энциклопедический словарь /Под ред. В. Н. Ярцевой, М.: Советская энциклопедия, 1990, 685 с.
20 Маннинг К., Рагхаван П., Шютце Ч. Введение в информационный поиск — М.: Вильямс, 2011
21 James Le. The 7 NLP Techniques That Will Change How You Communicate in the Future (Part I). Heartbeat, 2018.

22 Ben Lutkevich. Natural Language Processing (NLP). Technical Features Writer.
23 Koray Tuğberk GÜBÜR. NLTK Lemmatization: How to Lemmatize Words with NLTK? Holistic SEO, 2021.
24 Rebecca Dridan and Stephan Oepen. Tokenization: Returning to a long solved problem — a survey, contrastive experiment, recommendations, and toolkit. Association for Computational Linguistics, 2012.
25 Sarica S, Luo J (2021) Stopwords in technical language processing. PLoS ONE, 2021.
26 Kerem Kargın. NLP: Tokenization, Stemming, Lemmatization and Part of Speech Tagging. MLearning.ai, 2021.
27 Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. Attention is all you need. In Advances in neural information processing systems (cc. 5998-6008), 2017.
28 Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018.
29 Bahdanau et al. Neural Machine Translation by Jointly Learning to Align and Translate, 2014.
30 Pennington, J., Socher, R., & Manning, C. D. GloVe: Global Vectors for Word Representation, 2014.
31 Mikolov et al. Word2Vec, 2013.
32 Мегапьютер Интеллидженс. Megaputer PolyAnalyst, 2020.
33 PolyAnalyst: Extracting Semantic Structure from Text Data. Megaputer Intelligence, 2018.
34 Radu Gheorghe. Elasticsearch for Natural Language Processing, 2015.
35 Fabio Pinheiro, Vineeth Mohan. Mastering Elasticsearch for NLP, 2019.
36 David Pilato. Applied Text Analysis with Elasticsearch, 2019.
37 Alem media monitoring, 2021.

```
import pandas as pd
import numpy as np
import re
import nltk
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sklearn.metrics as metrics
from numpy import array
from numpy import asarray
from numpy import zeros
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from numpy import array
import keras
from keras.preprocessing.text import one_hot
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers.core import Activation, Dropout, Dense
from keras.models import Model
from keras.layers import Input
from keras.layers import Dense
from keras.layers import Flatten
from keras.layers import Conv1D
from keras.layers import LSTM
from keras.layers import GlobalMaxPooling1D
from keras.layers.convolutional import Conv1D
from keras.layers.convolutional import MaxPooling1D
from keras.layers.merge import concatenate
from keras.layers.embeddings import Embedding
from sklearn.model_selection import train_test_split
from keras.preprocessing.text import Tokenizer

Topics_rus = pd.read_csv("Topics_rus_balanced.csv")
Topics_kaz = pd.read_csv("Topics_kaz_balanced.csv")
Comments_rus = pd.read_csv("Comments_rus_balanced.csv")
Comments_kaz = pd.read_csv("Comments_kaz_balanced.csv")

Topics_rus
Topics_kaz.shape
Topics_rus.shape
Comments_rus.shape
```

```
Comments_kaz.shape

import seaborn as sns
sns.countplot(x='tonal_id', data=Topics_rus)

Topics_rus['text'] = Topics_rus['text'].apply(lambda x: x.lower())
Topics_kaz['text'] = Topics_kaz['text'].apply(lambda x: x.lower())
Comments_rus['text'] = Comments_rus['text'].apply(lambda x: x.lower())
Comments_kaz['text'] = Comments_kaz['text'].apply(lambda x: x.lower())

stop_words_russian = set(stopwords.words('russian'))
stop_words_kazakh = set(stopwords.words('kazakh'))

words_to_remove = []
for i in stop_words_russian:
    if i=='не' or i=='ни':
        words_to_remove.append(i)

words_to_remove2 = []
for i in stop_words_kazakh:
    if i=='жоқ' or i=='емес':
        words_to_remove2.append(i)

for i in words_to_remove:
    stop_words_russian.remove(i)

stop_words_russian

for i in words_to_remove2:
    stop_words_kazakh.remove(i)

stop_words_kazakh

for index, row in Topics_rus.iterrows():
    row['text'] = re.sub('[^a-zA-Za-яА-ЯӘІҢҒҮҰҚӨәіңғүұқөh]',' ',row['text'])
    tokens = word_tokenize(row['text'])
    tokens2 = [t for t in tokens if t not in stop_words_russian]
    Topics_rus.loc[index, 'text'] = ' '.join(tokens2)


for index, row in Comments_rus.iterrows():
    row['text'] = re.sub('[^a-zA-Za-яА-ЯӘІҢҒҮҰҚӨәіңғүұқөh]',' ',row['text'])
    tokens = word_tokenize(row['text'])
    tokens2 = [t for t in tokens if t not in stop_words_russian]
    Comments_rus.loc[index, 'text'] = ' '.join(tokens2)
```

```
Topics_rus[Topics_rus['tonal_id']==1]

for index, row in Topics_kaz.iterrows():
    row['text'] = re.sub('[^a-zA-Za-яА-ЯӘІҢҒҮҰҚӨәіңғүұқөh]',' ',row['text'])
    tokens = word_tokenize(row['text'])
    tokens2 = [t for t in tokens if t not in stop_words_kazakh]
    Topics_kaz.loc[index, 'text'] = ' '.join(tokens2)

for index, row in Comments_kaz.iterrows():
    row['text'] = re.sub('[^a-zA-Za-яА-ЯӘІҢҒҮҰҚӨәіңғүұқөh]',' ',row['text'])
    tokens = word_tokenize(row['text'])
    tokens2 = [t for t in tokens if t not in stop_words_kazakh]
    Comments_kaz.loc[index, 'text'] = ' '.join(tokens2)

Topics_kaz[Topics_kaz['tonal_id'] == 1]

x_topics_train_rus, x_topics_test_rus, y_topics_train_rus, y_topics_test_rus = train_test_split(Topics_rus['cleaned_text'], Topics_rus['tonal_id'], test_size=0.2, random_state=50)

x_topics_train_kaz, x_topics_test_kaz, y_topics_train_kaz, y_topics_test_kaz = train_test_split(Topics_kaz['cleaned_text'], Topics_kaz['tonal_id'], test_size=0.2, random_state=50)

x_comments_train_rus, x_comments_test_rus, y_comments_train_rus, y_comments_test_rus = train_test_split(Comments_rus['cleaned_text'], Comments_rus['tonal_id'], test_size=0.2, random_state=50)

x_comments_train_kaz, x_comments_test_kaz, y_comments_train_kaz, y_comments_test_kaz = train_test_split(Comments_kaz['cleaned_text'], Comments_kaz['tonal_id'], test_size=0.2, random_state=50)

import fasttext
import fasttext.util

ft = fasttext.load_model('cc.ru.300.bin')
ft2 = fasttext.load_model('cc.kk.300.bin')

tokenizer = Tokenizer(num_words=20000)
tokenizer.fit_on_texts(x_topics_train_rus)
x_topics_train_rus = tokenizer.texts_to_sequences(x_topics_train_rus)
x_topics_test_rus = tokenizer.texts_to_sequences(x_topics_test_rus)

tokenizer2 = Tokenizer(num_words=20000)
tokenizer2.fit_on_texts(x_topics_train_kaz)
```

```
x_topics_train_kaz = tokenizer.texts_to_sequences(x_topics_train_kaz)
x_topics_test_kaz = tokenizer.texts_to_sequences(x_topics_test_kaz)

tokenizer3 = Tokenizer(num_words=20000)
tokenizer3.fit_on_texts(x_comments_train_rus)
x_comments_train_rus = tokenizer3.texts_to_sequences(x_comments_train_rus)
x_comments_test_rus = tokenizer3.texts_to_sequences(x_comments_test_rus)

tokenizer4 = Tokenizer(num_words=20000)
tokenizer4.fit_on_texts(x_comments_train_kaz)
x_comments_train_kaz = tokenizer4.texts_to_sequences(x_comments_train_kaz)
x_comments_test_kaz = tokenizer4.texts_to_sequences(x_comments_test_kaz)

vocab_size = len(tokenizer.word_index) + 1
vocab_size2 = len(tokenizer2.word_index) + 1
vocab_size3 = len(tokenizer3.word_index) + 1
vocab_size4 = len(tokenizer4.word_index) + 1
maxlen = 300

x_topics_train_rus = pad_sequences(x_topics_train_rus, padding='post',
maxlen=maxlen)
x_topics_test_rus = pad_sequences(x_topics_test_rus, padding='post',
maxlen=maxlen)

x_topics_train_kaz = pad_sequences(x_topics_train_kaz, padding='post',
maxlen=maxlen)
x_topics_test_kaz = pad_sequences(x_topics_test_kaz, padding='post',
maxlen=maxlen)

x_comments_train_rus = pad_sequences(x_comments_train_rus, padding='post',
maxlen=maxlen)
x_comments_test_rus = pad_sequences(x_comments_test_rus, padding='post',
maxlen=maxlen)

x_comments_train_kaz = pad_sequences(x_comments_train_kaz, padding='post',
maxlen=maxlen)
x_comments_test_kaz = pad_sequences(x_comments_test_kaz, padding='post',
maxlen=maxlen)

embedding_matrix = zeros((vocab_size, 300))

embedding_matrix

embedding_matrix2 = zeros((vocab_size2, 300))
```

```
embedding_matrix2

embedding_matrix3 = zeros((vocab_size3, 300))

embedding_matrix3

embedding_matrix4 = zeros((vocab_size4, 300))

embedding_matrix4

for word, index in tokenizer.word_index.items():
    embedding_vector = ft.get_word_vector(word)
    if embedding_vector is not None:
        embedding_matrix[index] = embedding_vector

for word, index in tokenizer2.word_index.items():
    embedding_vector2 = ft2.get_word_vector(word)
    if embedding_vector2 is not None:
        embedding_matrix2[index] = embedding_vector2

for word, index in tokenizer3.word_index.items():
    embedding_vector3 = ft.get_word_vector(word)
    if embedding_vector3 is not None:
        embedding_matrix3[index] = embedding_vector3

for word, index in tokenizer4.word_index.items():
    embedding_vector4 = ft.get_word_vector(word)
    if embedding_vector4 is not None:
        embedding_matrix4[index] = embedding_vector4

import keras_metrics
import tensorflow as tf

length = len(x_topics_train_rus)
model = Sequential()
embedding_layer = Embedding(vocab_size, 300, weights=[embedding_matrix],
input_length=maxlen)
model.add(embedding_layer)
model.add(Conv1D(128, 5, activation='relu'))
model.add(GlobalMaxPooling1D())
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))

model = Sequential()
```

```
embedding_layer = Embedding(vocab_size, 300, weights=[embedding_matrix],
input_length=maxlen)
model.add(embedding_layer)
model.add(LSTM(16, return_sequences=True, dropout=0.3, recurrent_dropout=0.2))
model.add(LSTM(16, dropout=0.3, recurrent_dropout=0.2))
model.add(Dense(10))
model.add(Dense(10))
model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy',
keras_metrics.precision(), keras_metrics.recall()])
history = model.fit(x_topics_train_rus, y_topics_train_rus, batch_size=64,
epochs=10, verbose=1, validation_split=0.2)

score = model.evaluate(x_topics_test_rus, y_topics_test_rus, verbose=1)

print("Russian topics classification score: ", score[0])
print("Russian topics classification accuracy: ", score[1])
print("Russian topics classification precision: ", score[2])
print("Russian topics classification recall: ", score[3])
print("Russian topics classification F-measure",
2*score[2]*score[3]/(score[2]+score[3]))

from sklearn.metrics import roc_curve
y_topics_pred_rus = model.predict(x_topics_test_rus).ravel()
fpr_topics_rus, tpr_topics_rus, thresholds_topics_rus = roc_curve(y_topics_test_rus,
y_topics_pred_rus)
roc_auc_topics_rus = metrics.auc(fpr_topics_rus, tpr_topics_rus)
plt.title("ROC")
plt.plot(fpr_topics_rus, tpr_topics_rus, 'green', label = 'ROC for Russian texts =
%0.2f' % roc_auc_topics_rus)
plt.legend(loc = 'lower right')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()

model2 = Sequential()
embedding_layer2 = Embedding(vocab_size2, 300, weights=[embedding_matrix2],
input_length=maxlen)
model2.add(embedding_layer2)
model2.add(Conv1D(128, 5, activation='relu'))
model2.add(GlobalMaxPooling1D())
model2.add(Dropout(0.5))
```

```
model2.add(Dense(1, activation='sigmoid'))

model2 = Sequential()
embedding_layer2 = Embedding(vocab_size2, 300, weights=[embedding_matrix2],
input_length=maxlen)
model2.add(embedding_layer2)
model2.add(LSTM(16, return_sequences=True, dropout=0.3,
recurrent_dropout=0.2))
model2.add(LSTM(16, dropout=0.3, recurrent_dropout=0.2))
model2.add(Dense(10))
model2.add(Dense(10))
model2.add(Dense(1, activation='sigmoid'))
model2.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy',
keras_metrics.precision(), keras_metrics.recall()])

history2 = model2.fit(x_topics_train_kaz, y_topics_train_kaz, batch_size=64,
epochs=10, verbose=1, validation_split=0.2)

score2 = model2.evaluate(x_topics_test_kaz, y_topics_test_kaz, verbose=1)

print("Kazakh topics classification score: ", score2[0])
print("Kazakh topics classification accuracy: ", score2[1])
print("Kazakh topics classification precision: ", score2[2])
print("Kazakh topics classification recall: ", score2[3])
print("Kazakh topics classification F-measure",
2*score2[2]*score2[3]/(score2[2]+score2[3]))

from sklearn.metrics import roc_curve
y_topics_pred_kaz = model.predict(x_topics_test_kaz).ravel()
fpr_topics_kaz, tpr_topics_kaz, thresholds_topics_kaz =
roc_curve(y_topics_test_kaz, y_topics_pred_kaz)
roc_auc_topics_kaz = metrics.auc(fpr_topics_kaz, tpr_topics_kaz)
plt.title("ROC")
plt.plot(fpr_topics_kaz, tpr_topics_kaz, 'red', label = 'ROC for Kazakh texts = %0.2f'
% roc_auc_topics_kaz)
plt.legend(loc = 'lower right')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```

```
import numpy as np
import pandas as pd

Rus_topics = pd.read_csv("Topics_rus_balanced.csv")

Rus_topics_pos = Rus_topics[Rus_topics['tonal_id'] == 1]
Rus_topics_pos

Rus_topics_neg = Rus_topics[Rus_topics['tonal_id'] == 0]
Rus_topics_neg

Rus_topics_pos = Rus_topics_pos[:520]
Topics_rus = pd.concat([Rus_topics_pos, Rus_topics_neg])
Topics_rus

Topics_rus.to_csv("Topics_rus_balanced.csv")

Kaz_topics = pd.read_csv("Topics_kaz_balanced.csv")
Kaz_topics

Kaz_topics_neg = Kaz_topics[Kaz_topics['tonal_id'] == 0]
Kaz_topics_neg

Kaz_topics_pos = Kaz_topics[Kaz_topics['tonal_id'] == 1]
Kaz_topics_pos

Kaz_topics_neg = Kaz_topics_neg[:217]
Kaz_topics_pos = Kaz_topics_pos[:217]

Kaz_topics_pos

Kaz_topics_neg

Topics_kaz = pd.concat([Kaz_topics_pos, Kaz_topics_neg])
Kaz_topics

Kaz_topics.to_csv("Topics_kaz_balanced.csv")
Rus_comments = pd.read_csv("Comments_rus_balanced.csv")

Rus_comments_pos = Rus_comments[Rus_comments['tonal_id'] == 1]
Rus_comments_neg = Rus_comments[Rus_comments['tonal_id'] == 0]
```

```
Rus_comments_pos

Rus_comments_neg

Rus_comments_neg = Rus_comments_neg[:3757]
Rus_comments = pd.concat([Rus_comments_pos, Rus_comments_neg])
Rus_comments

Rus_comments.to_csv("Comments_rus_balanced.csv")
Kaz_comments = pd.read_csv("Comments_kaz_balanced.csv")

Kaz_comments
Kaz_comments_neg = Kaz_comments[Kaz_comments['tonal_id'] == 0]
Kaz_comments_pos = Kaz_comments[Kaz_comments['tonal_id'] == 1]
Kaz_comments_neg

Kaz_comments_pos = Kaz_comments_pos[:173]

Kaz_comments_pos
Kaz_comments_neg

Comments_kaz = pd.concat([Kaz_comments_pos, Kaz_comments_neg])
Comments_kaz
Comments_kaz.to_csv("Comments_kaz_balanced.csv")
```