МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Кафедра «Прикладная математика и информатика»
(наименование)

01.03.02 Прикладная математика и информатика
(код и наименование направления подготовки / специальности)

Компьютерные технологии и математическое моделирование
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Исследование и реализация алгоритмов анализа новостей в информационном пространстве

Обучающийся Н.Е. Санин
(Инициалы Фамилия) (личная подпись)

Руководитель
д.т.н., доцент, С.В. Мкртычев
(ученая степень(при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант О.А. Головач
(ученая степень(при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

Аннотация

Тема бакалаврской работы: «Исследование и реализация алгоритмов анализа новостей в информационном пространстве».

Бакалаврская работа состоит из введения, 3 глав, заключения, таблиц, списка литературы, включающего иностранные источники, и графической части на 23 листах.

Ключевой вопрос бакалаврской работы – разработка и внедрение алгоритмов анализа новостей в информационном пространстве. Мы затрагиваем проблему эффективной обработки новостных данных для извлечения значимых инсайтов, а также проблемы обеспечения точного и беспристрастного анализа в условиях быстро меняющейся информационной среды.

Цель работы – предоставить информацию о разработке исследовательских и внедренческих методологий для алгоритмов анализа новостей с целью улучшения понимания и обработки новостных данных. Это демонстрируется на примере разработки и внедрения конкретных алгоритмов, направленных на повышение точности и эффективности анализа новостей.

Выпускная работа может быть разделена на несколько логически связанных частей: анализ существующих алгоритмов анализа новостей; обоснование необходимости усовершенствованных методов анализа новостей; выбор подходящих алгоритмических подходов; внедрение и тестирование алгоритмов; технологические и проектные решения; пример зарубежного опыта в реализации передовых систем анализа новостей.

В заключение хотелось бы подчеркнуть, что данная работа актуальна для решения проблемы улучшения анализа новостей в информационном пространстве, а аналогичные технологические и конструктивные решения могут быть использованы для оптимизации анализа новостей в различных странах.

Abstract

The title of the graduation work is Research and implementation of news analysis algorithms in the information space.

The senior paper consists of an introduction, 3 parts, a conclusion, tables, a list of references including foreign sources, and a graphic part on 23 sheets.

The key issue of the thesis is the development and implementation of algorithms for analyzing news in the information space. We address the problem of effectively processing news data to extract meaningful insights, as well as the challenges of ensuring accurate and unbiased analysis in a rapidly changing information environment.

The aim of the work is to provide information about the development of research and implementation methodologies for news analysis algorithms to enhance the understanding and processing of news data. This is demonstrated through the design and implementation of specific algorithms aimed at improving the accuracy and efficiency of news analysis.

The graduation work may be divided into several logically connected parts which are: analysis of existing news analysis algorithms; justification of the need for enhanced news analysis methods; selection of appropriate algorithmic approaches; implementation and testing of the algorithms; technological and design solutions; an example of foreign experience in the implementation of advanced news analysis systems.

Finally, we present the work on successful foreign experiences of news analysis using advanced algorithms, which have significantly improved the accuracy and efficiency of analysis and enhanced the timeliness of information processing.

In conclusion, we would like to stress that this work is relevant for solving the problem of improving news analysis in the information space. Besides similar technological and constructive solutions can be used to optimize news analysis in various countries.

Оглавление

Введение

В прогрессивном мире необходимо наиболее быстро узнавать о последних событиях и трендах. И с развитием технологий способы получения новостей изменились. Больше не нужно ждать утренней газеты или следить за вечерними новостями по телевизору. Теперь люди могут узнавать обо всем новом с помощью интернета и социальных сетей всего лишь одним нажатием кнопки [5].

После этих изменений в том, как мы получаем новости, стало намного больше источников новостей в интернете [16]. В интернете есть много разных мест, откуда можно узнать новости, но не все они хорошие. Некоторые могут быть не очень точными или устаревшими. В такой куче информации может быть сложно найти самые свежие и правдивые новости [11].

Это важная тема, потому что сейчас важно быстро собирать и анализировать новости в онлайн-мире. Из-за большого объема информации в интернете программы могут помогать журналистам отслеживать последние события. Оценка популярности и влиятельности новостей тоже может помочь понять, какие истории важны, что полезно журналистам и экспертам по медиа. Вместо использования компьютерных программ для отслеживания новостей и составления рейтинга популярных тем можно делать это вручную. Но такой подход требует много времени и может содержать ошибки, потому что он зависит от большого количества работы вручную и ее интерпретации. При использовании ручных методов может случиться так, что мы выберем новости субъективно, основываясь на наших собственных предпочтениях и мнениях. Но если мы воспользуемся программным обеспечением для сбора новостей и создания рейтинга, мы сможем избежать таких ошибок и предвзятости. Это поможет дать наиболее верное и объективное представление о происходящем.

Объектом исследования бакалаврской работы является анализ новостей в информационном пространстве.

Предмет исследования – алгоритмы анализа новостей в информационном пространстве.

Цель работы – исследование и реализация алгоритмов анализа новостей в информационном пространстве.

В бакалаврской работе исследуется тематическое моделирование. Главная цель - изучить и реализация алгоритмов анализа новостей в информационном пространстве с помощью тематического моделирования, а также создать и протестировать программу для их использования.

Для достижения данной цели нужно выполнить следующие задачи:

- поставить задачу исследования и изучить методы тематического моделирования;
- проанализировать алгоритмы тематического моделирования;
- выполнить реализацию алгоритмов и протестировать разработанную программу.

Работа состоит из введения, трех глав, заключения и списка использованной литературы.

В первой главе описывается задача исследования и анализ методов тематического моделирования. Во второй главе проводится исследование и разбор алгоритмов анализа новостей с помощью тематического моделирования. В третьей главе рассматривается программная реализация и тестирование алгоритмов тематического моделирования. В заключении производятся выводы о проделанной работе.

Бакалаврская работа состоит из 63 страниц текста, 23 рисунков, 5 таблиц и 31 источника.

Глава 1 Описание сбора новостной информации

1.1 Постановка задачи и описание существующих методов и средств сбора новостной информации

Задачей данной работы является создание компьютерной программы, которая будет брать новости из различных мест и использовать их для того, чтобы понять, о чем сейчас говорят более всего люди. Программа будет набирать и изучать информацию из новостей, соцсетей и иных источников, чтобы определить свежие темы и узнать, в зависимости от активности людей в информационном пространстве, что сейчас больше популярно в обсуждении.

Цель этой работы – выработать достаточный и простой в использовании инструмент, чтобы люди и фирмы могли постоянно знать, что происходит в мире.

В зависимости от того, что нам необходимо узнать, мы можем подбирать различные методы брать новости. Например, ради извлечения новых новостей в настоящем времени мы можем использовать RSS-каналы [3],[28], которые автоматически обновляют новости. Для того чтобы вернее понять, что случается в мире, мы можем использовать особые сервисы, например, MediaCloud и EventRegistry [29]. Эти платформы также позволяют собирать данные из различных источников и анализировать их, чтобы выявить текущие важные события и тренды.

Чтобы узнать, что происходит в социальных сетях, используйте инструменты Social Searcher или Brandwatch, которые предоставляют аналитику по упоминаниям и тенденциям в социальных медиа, мы можем использовать особые программные инструменты, именуемые API, которые предоставляют сведения о постах, хэштегах и иных связанных событиях. Также существуют специализированные программы, как Hootsuite и Sprout Social, которые ориентируют проверять новости в социальных сетях [10],[14],[31].

Этот раздел будет о том, как мы можем собирать новости с помощью компьютерных технологий. В нем мы рассмотрим такие способы, как отслеживание социальных сетей, использование RSS-каналов и методы, позволяющие извлекать информацию прямо из интернет-страниц:

- в этой части мы обсудим, с чем сталкиваются журналисты, когда собирают новости. Это включает в себя то, насколько информация точна, как быстро её можно собрать и насколько можно доверять источникам;
- мы рассмотрим, как работают уже существующие системы, которые оценивают и ранжируют новости. В этой части мы обсудим алгоритм Google News, систему Alexa и NewsWhip Spike.

Тема работы будет основана на информации о том, что мы узнали в этой части.

Программа будет собирать новости как обычно, так и с помощью компьютерных технологий, чтобы создать большую базу данных новостей. Затем она будет использовать эту базу данных, чтобы показать, о чём больше всего говорят в новостях. Оценка будет учитывать сколько раз упоминается каждая история, какой тон общается о ней и где она была опубликована [9]. Программа будет проста в настройке, что позволит новостным организациям изменять систему ранжирования под свои нужды [1]. В целом, это приложение станет полезным инструментом для сбора и анализа новостей для медиакомпаний.

1.2 Методы поиска информации в Интернете

Существует несколько ключевых методов поиска информации в интернете. В зависимости от того, что ищет человек и зачем, эти способы могут использоваться по-разному или вместе:

- использование поисковиков в сети интернет;
- использование браузера для поиска информации;

– изучение новых источников.

Можно выделить несколько основных способов поиска информации в интернете, которые могут быть полезны для алгоритмов тематического моделирования.

Использование поисковых систем: Поисковые системы, такие как Google, Bing и другие, являются основным инструментом для нахождения информации в интернете. Они позволяют быстро находить релевантные источники по ключевым словам и фразам, что особенно полезно на начальных этапах исследования.

Применение браузера для поиска информации: Браузеры предоставляют различные инструменты для поиска информации, включая встроенные поисковые функции и возможности использования закладок и истории поиска для организации и повторного доступа к важной информации.

Изучение новых источников: Это включает в себя поиск новых, ранее неизвестных источников информации, таких как специализированные веб-сайты, форумы, блоги и научные публикации. Изучение новых источников может помочь в обнаружении уникальных данных и точек зрения, которые могут не быть доступны через стандартные поисковые системы.

Использование онлайн-библиотек и баз данных: Специализированные онлайн-библиотеки и базы данных, такие как Google Scholar, JSTOR, и PubMed, предлагают доступ к научным статьям, книгам и другим академическим ресурсам, которые могут быть чрезвычайно полезны для глубинного анализа и получения достоверных данных [27].

Социальные сети и онлайн-сообщества: Социальные сети (например, Twitter, LinkedIn) и специализированные онлайн-сообщества (например, Reddit, Quora) могут служить источниками актуальной информации и мнений экспертов. Участие в таких платформах позволяет получать данные от широкого круга пользователей и экспертов [20].

Веб-скрейпинг: автоматизированные методы извлечения данных с веб-сайтов с использованием инструментов веб-скрейпинга (например,

BeautifulSoup, Scrapy) могут существенно ускорить процесс сбора большого объема данных [17]. Этот подход особенно полезен, когда необходимо собрать данные с многочисленных веб-страниц быстро и эффективно.

Таким образом, сочетание автоматических и ручных методов поиска информации может существенно повысить качество сбора данных для алгоритмов тематического моделирования, обеспечивая более полное и точное моделирование тем.

Поскольку все сайты в интернете связаны между собой, мы можем собирать информацию, просматривая страницы одну за другой с помощью браузера. Хотя это делается вручную и кажется устаревшим способом в сети, которая включает миллионы сайтов, иногда ручной просмотр веб-страниц остается единственным способом получить нужные данные, особенно в завершающих этапах поиска, когда автоматический поиск не всегда помогает [22]. К этому типу поиска относится использование каталогов, списков по темам и небольших справочников [23].

1.3 Анализ существующих подходов к сбору и анализу новостной информации

Сбор и анализ новостей крайне важны в современном мире, где СМИ имеют большое влияние на формирование общественного мнения. Есть различные методы брать и анализировать новости: ручные и автоматические [26].

Ручные способы подключают действия человека, когда люди сами берут новости, разбирая газеты, глядя новости по телевизору, беря интервью или используя соцсети [30]. Эти способы занимают немало времени, и насколько четкая информация будет собрана, зависит от того, как опытен человек. Кроме того, при таком раскладе непросто обхватить все источники, и необходимые новости могут быть упущены.

Автоматизированные способы – это когда компьютерная программа

берет новости с различных мест в интернете, таких как новостные сайты, социальные сети и RSS-каналы [19]. Автоматизированные способы функционируют скоро и эффективно, они могут брать большое число информации из разных источников. Кроме того, подобные способы дозволяют анализировать сконцентрированные данные и находить в них направленности и закономерности, что могут быть упущены при применении ручных методов.

Один из методов анализа новостей – это анализ настроений. Этот подход является важной частью разработки программного обеспечения для сбора новостей и выявления самых обсуждаемых тем. В этом ходе анализируется язык, используемый в новостях, чтобы понять, как люди относятся к поставленной теме. С развитием цифровых новостей анализ настроений стал необходимым для компаний, правительств и частных лиц, чтобы находиться в курсе происшествий и обходить новости [17]. В контексте сбора новостей и анализа направленностей анализ настроений сможет посодействовать обнаружить свежие темы и тенденции, изучая, какие эмоции и расположения подают новостные статьи. Если смотреть за настроением в новостях, возможно увидеть, что волнует или интересует людей, и предсказать, что будет значительным в ближайшем будущем. Эта информация полезна для принятия решений, менеджмента и прочих стратегий бизнеса [21].

Для анализа настроений имеются различные инструменты и методы, такие как алгоритмы обработки естественного языка (NLP) и методы машинного обучения. Они помогают обнаруживать совместные черты в языке новостей и определять общее настроение статьи, будь то позитивное, негативное или нейтральное.

Одно из преимуществ использования анализа настроений в программе для анализа новостей – вероятность автоматически обнаруживать свежие тенденции. С помощью машинного обучения программа сможет сама выучиться различать всеобщие черты языка новостей и автоматически определять, что вызывает глубокий интерес или беспокойство.

В общем, анализ настроений бесконечно большой инструментарий для

создания программ, которые берут новости и определяют самые популярные темы. Если изучить, какой язык применяется в новостях, можно разузнать о свежих направленностях и быть спереди других в новостном цикле. По мере того, как интернет-новости становятся все популярнее, анализ настроений делается еще более необходимым для компаний, государственных систем и обычных людей, которые хотят находиться в курсе событий и обходить конкурентов [21].

Другим методом прибывает тематическое моделирование – это популярный метод, используемый в программном обеспечении для сбора новостей и определения самых актуальных тем. Он включает в себя выявление и изучение основных тем, присутствующих в обширных массивах данных, таких как статьи в новостях, публикации в социальных сетях и другой интернет-контент. Используя этот метод, программное обеспечение может быстро и точно определять и ранжировать наиболее важные и актуальные новости.

Использование тематического моделирования имеет значительное преимущество: оно позволяет извлечь более полное понимание новостей и тенденций. Вместо элементарного подсчета упоминаний темы либо слова, программа может распределить определенные темы и подтемы, которые более актуальны и важны. Это помогает людям вернее понимать смысл новостей и принимать более осмысленные заключения на их основе.

Кроме того, тематическое моделирование возможно использовать для выявления изменений и тенденций в течение времени. Анализируя наиболее популярные темы и подтемы в новостях, программа способна выявлять новые тенденции и изменения в общественном мнении [7],[24]. Это особенно полезно для компаний и других организаций, которым важно отслеживать последние события в своей отрасли или на рынке.

Разработка программного обеспечения для отслеживания новостей и выявления основных тенденций – это важное направление исследований. С помощью методов, таких как тематическое моделирование, разработчики

делают сильные инструменты, которые помогают людям быть в курсе событий, получать продуманные решения и быть впереди в быстро меняющемся мире [12]. Технологическая обработка и анализ информации включает процедуры, представленные на рисунок 1.

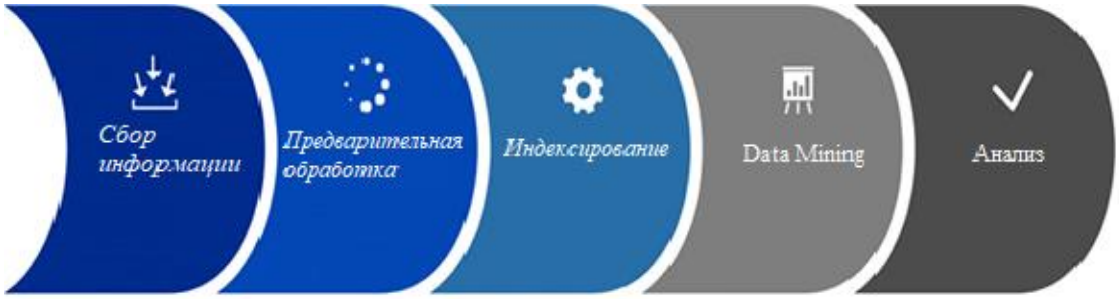

Рисунок 1 – Процедура обработки информации

При создании программы для отслеживания новостей и определения особенно функциональных тенденций возможно пользоваться автоматизированными методами, такими как анализ настроений и тематическое моделирование. Они помогут брать и анализировать информацию с разных источников, таковых как новостные сайты, социальные сети и RSS-каналы. После этого программа сможет употреблять методы машинного обучения для анализа информации и раскрытия свежих тенденций. Также она может расценивать особенно функциональные тенденции на базе скопленных и проанализированных данных.

Собирать и анализировать новости бесконечно важно в сегодняшнем мире, и для этого есть различные методы [2]. Автоматические способы быстрые, действенные и могут обрабатывать множество информации, что положительно для сбора и анализа новостей. Использование специфического программного обеспечения сможет помочь людям в информационном пространстве, маркетинге и политике обнаруживать актуальные события и основывать стратегии для действий.

1.4 Сравнительный анализ различных методов сбора и анализа данных

Сравнивать различные методы анализа и сбора актуальных данных для создания программы, которая будет собирать новости и оценивать активные тенденции, – это важная часть работы. При этом нужно обратить внимание на несколько вещей [18].

Способы, как мы получаем данные, могут быть разными: от сканирования веб-страниц до использования открытых программных интерфейсов (API) для сбора информации. Когда мы сравниваем эти способы, важно учитывать, насколько быстро данные собираются, их точность, методы сбора и доступность источников.

Перед тем как анализировать данные, их нужно подготовить. Это включает такие шаги, как лемматизация и стемминг. При сравнении методов сбора данных важно учитывать, насколько точно они обрабатывают данные перед анализом. Ведь точность обработки влияет на точность анализа данных.

При изучении данных важно смотреть на их распределение и настроение. Частотный анализ помогает узнать, как часто в тексте встречаются определенные слова и термины. Анализ тональности помогает определить, какое настроение несет текст. Когда выбираешь методы анализа, важно учитывать, насколько легко можно показать результаты в графическом виде, а также насколько точно и полно будут анализироваться данные.

Когда проводится сравнительный метод визуализации, важно понять, насколько просто и понятно он показывают результаты сбора и анализа данных. Также нужно учитывать, насколько сложно создать и поддерживать эти визуализации.

При сравнительном методе, важно понять, какие из них более выгодны с экономической точки зрения. Это означает учитывать все затраты на создание, поддержку, обновление и улучшение системы.

После сравнительного метода нужно выбрать самые подходящие для

задачи. Это можно оценить по таким критериям, как доступность, точность, скорость сбора и обработки данных, возможность визуализации результатов и экономическая эффективность.

Есть несколько способов собирать и анализировать данные для создания программного обеспечения, которое следит за актуальными новостями и трендами. Это включает в себя методы, такие как Web-скраппинг, анализ данных и обработку естественного языка. Давайте сравним эти методы, чтобы понять, какой из них лучше подходит для нашего проекта:

- web-скрапинг – это когда мы используем программы или скрипты для того, чтобы вытащить информацию с веб-сайтов. В нашем случае, это помогает нам извлекать новостные статьи из разных мест в интернете. Этот метод прост в использовании и быстро даёт много информации. Но некоторые изменения на сайте могут повлиять на то, как работает скрапинг, и он иногда не собирает всю нужную информацию;
- интеллектуальный анализ данных – это когда мы ищем закономерности и тенденции в больших объемах данных. В нашем случае, это помогает нам находить самые активные темы, анализируя, как часто встречаются ключевые слова в новостных статьях. Один из плюсов интеллектуального анализа данных в том, что он может раскрывать информацию, которая не так просто заметить. Но для его работы нужно много вычислительной мощности;
- обработка естественного языка (НЛП) – это когда мы анализируем тексты, чтобы понять их смысл. В нашем случае, НЛП помогает определять настроение новостных статей и разделять их по темам. Один из плюсов НЛП в том, что он разрешает постигать новости более глубоко, нежели простой анализ главных слов. Но для того чтобы НЛП работало хорошо, необходимо множество обучающих данных.

Каждый из данных способов имеет свои плюсы и минусы, и выбор, какой способ использовать, зависит от того, что необходимо для проекта. Web-скраппинг хорош, если главны скорость и простота, а интеллектуальный анализ информации и обработка естественного языка скорее идут для проектов, что требуют более полного понимания [13]. В итоге, применение всех данных способов совместно сможет являться наилучшим методом для того, чтобы брать и анализировать новости качественно и точно. В таблице 1 представлены методы сбора и анализа данных.

Таблица 1 – Сравнительный анализ методов анализа новостей в информационном пространстве

Методы	Преимущества	Недостатки
Web-скрейпинг	Позволяет быстро собирать большие объемы данных из различных источников.	Может нарушать правила использования сайтов, требует обработки больших данных
Мониторинг социальных сетей	Обеспечивает доступ к актуальным и разнообразным данным, позволяет отслеживать тренды и мнения.	Данные могут быть шумными и нерепрезентативными, сложности с приватностью.
Обзорное исследование	Предоставляет широкое понимание темы, основываясь на уже существующей литературе и данных.	Может быть ограничено доступностью и качеством существующих источников.
Интеллектуальный анализ данных	Помогает выявлять скрытые паттерны и тренды, повышает точность анализа.	Требует значительных вычислительных ресурсов и специализированных знаний.
Машинное обучение	Автоматизирует анализ больших данных, улучшает прогнозирование и классификацию.	Необходимость в больших обучающих выборках, сложность в интерпретации моделей.

Тематическое моделирование помогает нам понять, какие темы присутствуют в наших текстах и насколько они сильны. Он ищет темы в текстах, не требуя заранее подготовленных данных о них. Это означает, что мы можем использовать тематическое моделирование, даже если у нас нет заранее размеченных текстов. Мы просто даем алгоритмам тексты, и они находят в них темы.

Для сравнения методов тематического моделирования используем таблицу 2.

Таблица 2 – Сравнение методов тематического моделирования

Методы	Преимущества	Недостатки
LSA	Может различать контексты, в которых используются многозначные слова, что помогает в понимании текста на более глубоком уровне. Уменьшает размерность данных, что упрощает их обработку и анализ Улучшает результаты поиска, учитывая скрытые семантические отношения между словами. Может автоматически выявлять темы и скрытые связи в текстах. эффективно работает с большими объемами текста, что делает его полезным для анализа больших данных	Результаты LSA могут быть трудными для интерпретации, так как скрытые компоненты не всегда очевидны. требует значительных вычислительных ресурсов, особенно для обработки больших корпусов текста. Результаты LSA сильно зависят от качества и объема исходных данных
NMF	Быстрая переработка огромных размеров информации в настоящем времени. сможет получать необходимые темы без предварительной информации либо понимания глубокого значения входных данных.	Временами подает ошибочные смысловые результаты.
LDA	Способен исследовать долгие тексты и выделять прилагательные и существительные в темах. действует с текстами различной длины, используя статистическую образец для выявления тайных тем.	Не способен имитировать связи посреди различными темами, что ориентируют осознать текстуру документов. метод функционирует медленно.
PLSA	Использует вероятностную образец для разбора текстовых данных, предполагая, что каждый документ включает немного сокрытых тем.	Нельзя проверять разреженность получаемых вероятностных матриц. сумма параметров PLSA вырастает линейно с повышением численности документов, что сможет повергнуть к переобучению модели.
ARTM	Позволяет использовать разные модификации и методы оценки, что разрешает доставать более четкие и полезные результаты.	ARTM представляется ресурсоемким процессом, спрашивающим наибольшего количества вычислительных ресурсов ради обрабатывания текстовых данных.

Каждый из данных методов тематического моделирования имеет свои преимущества и недостатки, потому существенно сопоставлять их перед подбором определенного метода.

Выводы по главе 1

В первой главе нашей работы мы поставили задачи исследования и осмотрели разные методы тематического моделирования. В результате нашего разбора мы сделали последующие выводы:

- выбор метода тематического моделирования зависит от того, что именно мы желаем добиться и какие у нас данные;
- метод LSA разыскивает совместные закономерности в матрице, используя сингулярное разложение. Он не предусматривает связь в тексте, что может являться его недостатком;
- метод NMF акцентирует темы в текстах, используя матричное разложение;
- метод PLSA точно имитирует текстовые данные, но может запрашивать немало времени на обрабатывание огромных объемов информации;
- метод LDA точно определяет тайные темы и хорошо действует с огромными размерами текста, однако может вызвать много времени для обработки;
- ARTM дозволяет объединять документы по темам, что упрощает исследование огромных объемов текста и помогает обнаруживать связи среди ними.

PLSA зачастую применяется для разбора огромных данных, таких как тексты, изображения и видео, и справляется с разными проблемами, такими как разреженность и шум. LDA отлично идет для огромных объемов информации и сможет действовать с текстами на разных языках. ARTM владеет высочайшей быстротой обработки, что позволяет действовать с большими размерами информации.

Глава 2 Анализ алгоритмов тематического моделирования

2.1 Обзор алгоритмов формирования рейтинга новостей

В мире, где информация постоянно меняется, важно быть в курсе последних событий. Однако это сложно из-за большого объема информации в интернете. В этой части мы рассмотрим методы, которые помогают узнавать, что сейчас в тренде.

Тенденция – это общее направление, в котором что-то развивается или меняется. В контексте информации это означает, что сейчас эта тема очень популярна и многие об этом говорят [1].

Рейтинг трендов – это список самых популярных тем, о которых много говорят в определенный момент времени. Этот список составляется на основе таких вещей, как сколько раз они упоминаются, сколько статей написано на эту тему и откуда идет информация.

Для составления списка популярных тем в новостях используются различные методы и алгоритмы:

- скрытое распределение Дирихле (LDA) – это метод, который помогает понять, о чем говорят тексты. Он выделяет группы слов, которые часто упоминаются вместе, и называет их темами. Например, если в текстах часто упоминаются слова "автомобиль", "дорога", "бензин", то можно предположить, что эти тексты связаны с темой "автомобили". LDA помогает определить, какие темы самые популярные в данном наборе текстов, что полезно для анализа трендов;
- алгоритм наивного Байеса – это способ определения того, к какой категории относится текст. Он анализирует слова в тексте и пытается определить, о чем этот текст. Например, если в тексте много слов, связанных с путешествиями, алгоритм скорее всего определит этот текст как относящийся к категории "путешествия". В контексте

рейтинга трендов, алгоритм помогает определить, какие темы наиболее популярны в текстах, что помогает понять, о чем сейчас много говорят.

Есть разные способы оценки популярности тем или явлений в новостях, и выбор зависит от контекста и доступных данных [4],[25]. Некоторые обычные методы включают следующие виды анализа:

- частотный анализ – это метод, что смотрит, сколько раз каждый элемент либо явление возникает в наборе данных. Это позволяет узнать, какие элементы либо действия самые общераспространенные либо зачастую встречающиеся, что может помочь осознать направленности и закономерности в данных. данный способ несложный и просто понятный, и он отлично функционирует с большими размерами данных. впрочем он сможет не быть подходящим для сложных данных с запутанными связями среди элементами, потому что не учитывает связи среди ними;
- анализ настроений – это метод, что применяет обрабатывание естественного языка (NLP) для нахождения чувств, соединенных с любой темой или типом, и составляет показатель по тому, как они положительные или отрицательные;
- алгоритм разбора тональности – это метод обрабатывания текста, который определяет, положительный, критический или нейтральный курс сообщений. Он изучает слова и фразы в тексте, чтобы понять, какое состояние они передают, и присваивает любой доли текста оценку. Для этого метод применяет разные методы, такие как машинное обучение и байесовский анализ. Он хорошо функционирует в настоящем времени, сможет обрабатывать огромные объемы информации с различных источников, таких как социальные сети, новости и отзывы;
- машинное обучение – это метод сбора данных, который позволяет компьютерным системам автоматически извлекать знания из опыта.

– алгоритмы ранжирования трендов – это как машины, которые обучаются с огромных комплектов данных, чтобы обнаруживать закономерности и упорядочивать тренды. Они обычно применяют методы, что помогают осмыслить информацию из неструктурированных данных, таких как тексты, иллюстрации и видео.

Алгоритм глядит на данные из разных мест и выявляет, какие слова и темы пересекутся плотнее всего. затем он упорядочивает данные тренды по тому, как они активны и популярны в течение времени.

Характеристики алгоритмов ранжирования трендов подключают способность замечать за свежими трендами в реальном времени, способность обрабатывать огромные размеры информации и гибкость в адаптации к изменениям в трендах и интересах пользователей. Их можно использовать в разных областях, таких как маркетинг, исследование социальных сетей и прогнозирование экономических трендов.

Мы можем анализировать тенденции и категории, используя методы обработки естественного языка (Natural Language Processing, NLP), такие как тематическое моделирование, которое выявляет основные темы и концепции в данных, или распознавание образов, которое обнаруживает повторяющиеся паттерны и структуры [8].

2.2 Описание методов сбора и анализа новостей

Есть несколько способов получения новостей, включая:

– интервью. Журналисты могут задавать вопросы и брать интервью у различных людей, таких как эксперты или свидетели, чтобы узнать больше о событиях или историях;
– исследование – процесс изучения статей, новостей и других материалов, чтобы собрать информацию о конкретной теме;

- пресс-релизы – специальная информация, которую организации отправляют в СМИ и журналистам, чтобы рассказать о своих новостях или событиях;
- социальные сети – особые места в интернете, где люди общаются друг с другом. Можно использовать эти места для получения информации, особенно от людей, которые видели события и обновления в режиме реального времени;
- пресс-конференции – когда организации собираются, чтобы рассказать журналистам о своих новостях и отвечать на их вопросы.;
- наблюдения – когда журналисты сами наблюдают за событиями или ситуациями, чтобы узнать больше об этой истории;
- исследование данных – когда журналисты собирают информацию и анализируют ее, чтобы найти закономерности или тенденции, которые делают истории интересными для публикации в газетах или журналах;
- пресс-агентства – организации, которые собирают и распространяют новости со всего мира. Журналисты могут подписаться на них, чтобы получать актуальную информацию.

Описание методов анализа данных:

После того, как данные собраны, нужно изучить их, чтобы определить, какие тенденции самые значимые.

Обработка естественного языка (НЛП) – это способ анализа текстов. Он помогает нам понять, что написано в текстах, и найти важные сведения. Например, мы можем использовать НЛП для изучения собранных новостей и определения самых частых контекстных слов и фраз.

Для приложения, которое собирает и анализирует информацию, оптимальным решением будет интеграция методов обработки естественного языка (NLP), анализа настроений и машинного обучения. NLP позволит извлекать и обрабатывать текстовую информацию. Анализ настроений поможет определить эмоциональный окрас текстов.

Алгоритмы машинного обучения отлично справляются с анализом обширных объемов данных, выявлением закономерностей и формулированием прогнозов на их основе. Если обучить такой алгоритм на данных о новостных статьях и их аудитории, он сможет выявить, какие статьи наиболее популярны, и упорядочить их соответственно.

Использование методов машинного обучения для анализа текста и оценки настроений в приложении позволяет автоматизировать и улучшить процесс принятия решений. Комбинация предварительной обработки текста, алгоритмов анализа настроений и обучающихся моделей обеспечивает точный и надежный анализ, который можно использовать в различных бизнес-процессах.

2.3 Способы хранения информации и выбор БД

Есть различные способы структурирования для новостной информации, так существует множество реляционных моделей баз данных:

- web модель базы данных – это способ хранения данных, где они организованы как связанный граф. Этот метод гибкий и позволяет хранить данные разной структуры. Однако он не всегда эффективен при выполнении операций обработки данных, поэтому не так широко распространен;
- иерархическая модель баз данных – это способ организации данных, где они представлены в виде дерева без циклов. Это значит, что каждый элемент данных связан с одним или несколькими родительскими элементами, но не может иметь циклических связей. Такая модель часто используется в каталогах, например, в файловых системах, где файлы и папки организованы в иерархическую структуру;
- реляционная модель баз данных – это способ организации данных, где они хранятся в виде таблиц, состоящих из строк и столбцов.

Каждая таблица содержит связанные данные, которые могут быть легко связаны между собой. Эта модель основана на математической теории множеств и используется повсеместно из-за своей эффективности при решении множества задач;

– объектно-ориентированная модель баз данных – это способ организации данных, где данные хранятся как объекты, которые могут взаимодействовать друг с другом. Эта модель соответствует методу проектирования информационных систем, называемому объектно-ориентированным моделированием. Хотя эта модель считается перспективной уже много лет, она все еще не так широко распространена, как реляционная модель баз данных.

Таблица 3 – Сравнительный анализ СУБД для хранения данных новостей

СУБД	Плюсы	Минусы
MySQL	Высокая производительность; Широкая поддержка сообщества; Простота в установке и настройке; Хорошо подходит для веб-приложений.	Ограниченная поддержка ACID в старых версиях; Ограниченная функциональность в сравнении с PostgreSQL.
Microsoft SQL сервер	Высокая производительность; Хорошая интеграция с продуктами Microsoft; Мощные аналитические и BI инструменты	Высокая цена лицензий; Работает оптимально в экосистеме Microsoft.
PostgreSQL	Полная поддержка ACID Поддержка сложных запросов и индексов; Расширяемость и гибкость. Сильная поддержка JSON	Сложнее в настройке и управлении; Потребляет больше ресурсов.

Проанализировав таблицу сравнения систем управления базами данных (СУБД), можно заключить, что PostgreSQL является оптимальным выбором для нас. Эта система с открытым исходным кодом отличается возможностью масштабирования, поддержкой сложных запросов и эффективным текстовым поиском и индексированием данных. Также PostgreSQL славится своей надежностью и безопасностью.

Тем не менее, окончательный выбор зависит от конкретных требований и ограничений нашего ПО для анализа новостной информации.

2.4 Описание тематических алгоритмов анализа новостей

Алгоритмы тематического моделирования отображают в документе скрытые темы и их распределение, а также выявляют ключевые слова, характерные для каждой темы. Основная цель этих алгоритмов структурировать и организовать большие объемы текстовой информации.

Темы, которые ищут данные алгоритмы, презентуют собой категории слов, где всякое слово сопряжено с темами в различной степени. Это означает, что одно и то же слово может быть частью многих тем с различной степенью ассоциации. В статистическом значении темы возможно анализировать как распределение слов.

Осмотрим модель pLSA. Имитация возможности p(w|d) появления слов w в документах d в модели pLSA осуществляется через соответствующую оценку вероятности на основе параметров модели. В модели pLSA, вероятность появления слова w в документе d может быть выражена как сумма произведений p(w|t) * p(t|d) по всем темам t, где p(w|t) – вероятность слова w в теме t, а p(t|d) – вероятность темы t в документе d. Эти вероятности могут быть вычислены на основе параметров модели, полученных в результате обучения с использованием EM-алгоритма. Таким образом, имитация возможности появления слов в документах в модели pLSA достигается путем вычисления вероятностей на основе обученных параметров. Тематическая модификация явления слов в документах смотрится следующим образом (1):

$$p(d, w) = p(d)p(w|d) = p(d) \sum_{t \in T} p(w|t)p(t|d), \quad (1)$$

где d–документ коллекции;

w–терм;

t–тема.

Эти вероятности являются основой для моделирования распределения слов в документах и тем в документах, соответственно. Они могут быть оценены с использованием EM-алгоритма в процессе обучения модели на основе наблюдаемых данных, поэтому следует поставить характеристики модификации φwt = p(w|t) и θtd = p(t|d). Ставится цель максимизации логарифма правдоподобия (2) [7]:

$$L(\Phi,\Theta) = \sum_{d\in D}\sum_{w\in W} n_{dw}\log p(d,w) = \sum_{d\in D}\sum_{w\in W} n_{dw}\log\sum_{t\in T}\varphi_{wt}\theta_{td} \to \max_{\Phi,\Theta},$$

$$\sum_{w\in W}\varphi_{wt} = 1, \varphi_{wt} \geq 0, \sum_{t\in T}\theta_{td} = 1, \theta_{td} \geq 0, \quad (2)$$

где Φ = (φwt)W×T;

Θ = (θtd)T×D;

ndw – количество вхождений терма w документ d.

В процессе решения задачи с применением EM-алгоритма в модели pLSA происходит итеративная оптимизация функции правдоподобия. Этот процесс включает два основных шага: E-шаг, на котором оцениваются скрытые переменные модели, и M-шаг, на котором обновляются параметры модели с целью максимизации правдоподобия. Эти шаги повторяются до сходимости алгоритма, когда изменения параметров становятся незначительными. Таким образом, EM-алгоритм в модели pLSA ищет локальный максимум логарифма правдоподобия, учитывая структуру модели и наблюдаемые данные.

Алгоритм решения задачи EM (Expectation-Maximization) состоит из двух основных шагов: E-шаг (Expectation) и M-шаг (Maximization).

На E-шаге происходит оценка параметров модели при фиксированных текущих значениях параметров. Это включает вычисление ожидаемых значений скрытых переменных для каждого наблюдения на основе текущей модели.

На М-шаге происходит обновление параметров модели на основе полученных на Е-шаге оценок скрытых переменных. Это включает максимизацию правдоподобия путем обновления параметров модели.

Эти два шага последовательно выполняются до достижения сходимости, то есть до тех пор, пока изменения параметров между итерациями не станут незначительными.

Алгоритм PLSA-EM (рисунок 2) представляет собой разумный метод ЕМ-алгоритма для модели PLSA: собираются документы D, задается количество тем |T|, исходные приближения Θ и Φ; вычисляются распределения Θ и Φ;

Шаг 1: Повторять до сходимости:

Шаг 2: Сбросить $\hat{n}_{wt}, \hat{n}_{dt}, \hat{n}_t, \hat{n}_d, \hat{n}_{dwt}$ для всех $d \in D, w \in W, t \in T$;

Шаг 3: Для всех $d \in D, w \in d$;

Шаг 4: $= \sum_{t \in T} \phi_{wt}\theta_{td}$;

Шаг 5: Для всех $t \in T$ таких, что $\phi_{wt}\theta_{td} > 0$;

Шаг 6: Увеличить $\hat{n}_{wt}, \hat{n}_{dt}, \hat{n}_t$ на $\delta = \frac{n_{dw}\phi_{wt}\theta_{td}}{Z}$;

Шаг 7: $= \frac{\hat{n}_{wt}}{\hat{n}_t}$ для всех $w \in W, t \in T$;

Шаг 8: $= \frac{\hat{n}_{dt}}{\hat{n}_d}$ для всех $d \in D, t \in T$;

Рисунок 2 – Алгоритм правдоподобия ЕМ рациональный

Алгоритм PLSA-GEM (рисунок 3) представляет собой разумный метод ЕМ-алгоритма для модели PLSA: собираются документы D, задается количество тем |T|, исходные приближения Θ и Φ; осуществляется итерация, где на каждом шаге выполняются Е-шаг для оценки скрытых переменных и М-шаг для обновления параметров модели;

Шаг 1: Обнулить $\hat{n}_{wt}$, $\hat{n}_{dt}$, $\hat{n}_t$, $\hat{n}_d$, $\hat{n}_{dwt}$ для всех $d \in D$, $w \in W$, $t \in T$;
Шаг 2: Пока не выполнится критерий остановки, повторять итерации:
Шаг 3: Для всех $d \in D$, $w \in d$;
Шаг 4: Вычислить $Z = \sum_{t \in T} \phi_{wt}\theta_{td}$;
Шаг 5: Для всех $t \in T$, таких что $n_{dwt} > 0$ или $\phi_{wt}\theta_{td} > 0$;
Шаг 6: Вычислить $\delta = \frac{n_{dw}\phi_{wt}\theta_{td}}{Z}$;
Шаг 7: Увеличить $\hat{n}_{wt}$, $\hat{n}_{dt}$, $\hat{n}_t$ на $(\delta - n_{dwt})$;
Шаг 8: Установить $n_{dwt} := \delta$;
Шаг 9: Когда нужно пересчитать параметры Φ и Θ:
Шаг 10: Установить $\phi_{wt} = \frac{\hat{n}_{wt}}{\hat{n}_t}$ для всех $w \in W$, $t \in T$, таких что $\hat{n}_{wt}$ изменился;
Шаг 11: Установить $\theta_{td} = \frac{\hat{n}_{dt}}{\hat{n}_d}$ для всех $d \in D$, $t \in T$, таких что $\hat{n}_{dt}$ изменился.

Рисунок 3 – Алгоритм PLSA-GEM

Обобщенный EM-алгоритм для PLSA/LDA отличается от обычного EM-алгоритма в использовании дополнительных шагов регуляризации, адаптивного обновления параметров, улучшенных методов оценки правдоподобия данных и расширенных методов анализа результатов. Эти особенности позволяют GEM достигать более высокой точности и устойчивости при обучении моделей PLSA/LDA.

В алгоритмах PLSA-EM и PLSA-GEM обновления параметров модели происходят на каждой итерации EM-алгоритма, когда выполняются E-шаг и M-шаг. На E-шаге оцениваются скрытые переменные модели, а на M-шаге обновляются параметры модели с целью максимизации ожидаемого правдоподобия. Эти шаги повторяются до сходимости алгоритма, когда изменения параметров между итерациями становятся незначительными.

В обновлении после любого термина, возможно, не сохранять значения φwt, θtd. Это может быть особенно полезным в случае, если параметры модели нужно обновить с нуля на каждой итерации или если размер данных слишком велик для хранения всех параметров. Однако, в большинстве случаев,

сохранение и использование предыдущих значений параметров может ускорить сходимость алгоритма и повысить его эффективность.

Недостатком метода PLSA-GEM прибывает потребность держать скопление значений n̂dwt ради любого термина (d,w).

Модель pLSA имеет множество параметров, и она может стать расположенной к переобучению, особенно если у нас множество тем и недостаточно информации для обучения. Решением данной трудности сможет составлять незначительное модифицирование в шаге E-шага EM-алгоритма.

Проблему отличия тематических и низкоприоритетных долей смеси тем в модели pLSA можно решить с помощью регуляризации путем введения дополнительных ограничений на параметры модели.

В частности, можно использовать различные формы регуляризации, такие как L1 или L2 регуляризация, чтобы накладывать штрафы на некоторые компоненты распределений тем. Например, можно наложить L1-регуляризацию на распределения слов в низкоприоритетных темах, чтобы сделать их более разреженными и менее влиятельными в общей смеси тем.

Также можно использовать структурную регуляризацию, которая учитывает связи между темами и документами, чтобы более четко различать тематические и низкоприоритетные доли смеси тем. Например, можно наложить ограничения на сумму весов тем в каждом документе, чтобы обеспечить, что низкоприоритетные темы не будут иметь значительного влияния на содержание документов.

Такие подходы к регуляризации помогут улучшить интерпретируемость и качество модели pLSA, позволяя более явно различать между тематическими и низкоприоритетными долями смеси тем.

Существует несколько способов разреживания распределений p(t|d,w) (вероятность темы t при условии документа d и слова w) через EM-алгоритм в модели PLSA (Probabilistic Latent Semantic Analysis). Один из них – это добавление априорных распределений на параметры модели, например, дирихле-распределений на Θ и Φ, чтобы сделать распределения более

разреженными. Другой способ – использование регуляризации, добавление слагаемых в функцию правдоподобия, например, L1 или L2 регуляризации, для стимулирования разреживания распределений. Также возможно применение методов разреженной матричной факторизации, методов сжатия данных, а также использование механизмов устойчивости к шумам, чтобы контролировать степень разреженности распределений и улучшить качество модели PLSA.

В современном мире способы регуляризации активно применяются в тематическом моделировании для улучшения качества и интерпретируемости полученных тематических моделей. Регуляризация может включать в себя использование L1 и L2 регуляризации для контроля за параметрами модели, структурную регуляризацию для управления структурой модели, применение топических априоров для отражения априорных знаний о тематической структуре данных, а также методы разреживания распределений тем для повышения их интерпретируемости. Автоматический подбор параметров регуляризации также важен для достижения оптимального баланса между переобучением и недообучением модели. В целом, регуляризация играет ключевую роль в создании более точных, интерпретируемых и обобщающих моделей на основе текстовых данных.

Теперь рассмотрим модель LDA. Это широко распространенная вероятностная производящая модель, используемая ради разбора тематической структуры в коллекции документов. как модели pLSA, добавление ограничений на вид распределений в модели LDA требует модификации M-шага для обеспечения их соблюдения. Это может включать использование метода множителей Лагранжа, применение проекции для строгих ограничений, использование специализированных оптимизационных методов или введение регуляризации в функцию правдоподобия. Каждый из этих подходов направлен на обеспечение соблюдения ограничений при обновлении параметров модели и на обеспечение стабильного и эффективного процесса обучения модели. Самая простая и популярная из них приведенная

ниже формула (3) [7].

$$\varphi_{wt} \infty n_{wt} + \beta_{w,}\theta_{td} \infty n_{td} + \alpha_t. \qquad (3)$$

Это соответствует регуляризатору сглаживания в подходе ARTM.

Скрытое распределение Дирихле – такая модель, которая помогает нам осмыслить связи среди разных документов в скопленной коллекции текстов. LDA применяет алгоритм Гиббса и перплексию, для оценки характеристики модели.

Алгоритм Гиббса в LDA – это способ оценки, который помогает понять, какие слова связаны с какими темами в документах. Он предполагает, что каждое слово в документе имеет тему. Это позволяет нам понять вероятности тем и слов в темах из наших данных (текстов).

Это позволяет нам лучше понять, как связаны разные темы между собой.

Скрытое распределение Дирихле (LDA) – это метод, который помогает понять, о чем говорят тексты. Он выделяет группы слов, которые часто упоминаются вместе, и называет их темами. Например, если в текстах часто упоминаются слова "автомобиль", "дорога", "бензин", то можно предположить, что эти тексты связаны с темой "автомобили". LDA помогает определить, какие темы самые популярные в данном наборе текстов, что полезно для анализа трендов. Модель LDA является одной из популярных тематических моделей, но не всегда является самой предпочтительной в реальных приложениях. Выбор модели зависит от конкретной задачи, требований к интерпретируемости результатов и характеристик данных. В ряде сценариев LDA может быть оптимальным выбором, особенно когда важна простота и интерпретируемость. Однако в других случаях могут быть предпочтительны другие модели, такие как ARTM, NMF, в зависимости от характеристик данных и требований к анализу. Таким образом, хотя LDA широко используется, она не всегда является единственным или наиболее популярным выбором в реальных приложениях. Она дает точные результаты и может быть

обучена онлайн. В модели LDA мы предполагаем, что каждый документ в нашей коллекции состоит из разных тем, и каждая тема представляет собой набор слов, которые часто встречаются вместе. Один из больших плюсов LDA заключается в том, что мы можем извлечь темы из коллекции документов без предварительного знания о том, о чем они.

Модель LDA (Latent Dirichlet Allocation) – одна из известных тематических моделей в обработке естественного языка. В области обработки текстов существует множество других методов и моделей, каждая из которых применима в зависимости от конкретной задачи. LDA широко применяется для извлечения скрытой структуры тем из текстовых данных, однако другие модели могут быть предпочтительны для различных задач, таких как классификация текста, извлечение информации и машинный перевод. Таким образом, хотя LDA популярна и широко используется, она не является единственным или всегда наиболее популярным выбором в обработке естественного языка.

Но создание комбинированных и многоцелевых тематических моделей остается сложной задачей из-за трудностей в выводе, особенно в случае использования предшествующего Дирихле. Это мало изучено в литературе.

Алгоритм наивного Байеса – это способ определения того, к какой категории относится текст. Он анализирует слова в тексте и пытается определить, о чем этот текст. Например, если в тексте много слов, связанных с путешествиями, алгоритм скорее всего определит этот текст как относящийся к категории "путешествия". В контексте рейтинга трендов, алгоритм помогает определить, какие темы наиболее популярны в текстах, что помогает понять, о чем сейчас много говорят.

Еще одна запутанность содержится в том, что предшествующий Дирихле не отвечает натуральным предположениям о разреженности. Обычно документ включает мало тем, и любая тема охватывает несколько слов. Поэтому, большинство слов и тем должны иметь низкие или нулевые вероятности встречи. Разреженность в анализе больших текстовых наборов

означает, что матрицы, используемые для представления данных, содержат много нулевых значений или значений, близких к нулю. Это происходит из-за того, что каждый документ содержит лишь небольшой набор слов из всего словаря, что приводит к тому, что большинство компонентов вектора документа равны нулю. Такая разреженность является типичной для текстовых данных из-за обширного словаря и разнообразия слов, использованных в текстах. Это создает особые вызовы для эффективной обработки и анализа таких данных, требуя специальных методов хранения и обработки разреженных матриц.

Поэтому, в ситуации разреженности в текстовых данных большинство текстов и тем будут иметь нулевые вероятности встречи. Это связано с тем, что каждый текст и каждая тема представлены в виде векторов, в которых каждая компонента соответствует вероятности встречи конкретного слова в тексте или темы. Поскольку тексты обычно содержат лишь ограниченный набор слов из всего словаря, большинство компонентов вектора будет равно нулю или близко к нулю. То же самое касается и тем: поскольку каждая тема представлена как распределение вероятностей по словам, большинство слов в данной теме будут иметь нулевую вероятность, если они не являются ключевыми характеристиками этой темы.

Для того, чтобы постановить эти проблемы, мы используем элементарный подход, именуемый Аддитивной Регуляризацией Тематических Моделей (ARTM). Если мы проходим тематическую модель на базе коллекции документов, мы сходимся с тяжелой проблемой – обнаружить множество разных решений.

ARTM – это метод моделирования тем, который использует регуляризацию для улучшения и контроля качества получаемых тематических моделей. Этот метод представляет собой расширение классических тематических моделей, таких как PLSA и LDA, путем добавления регуляризаторов. Регуляризация позволяет вводить дополнительные ограничения на параметры модели, такие как сглаживание, разреживание и

взаимодействие, для предотвращения переобучения и улучшения интерпретируемости полученных тем. Это делает ARTM популярным методом в анализе текстовых данных, так как он обеспечивает более устойчивые и информативные тематические модели.

Аддитивная регуляризация различается от байесовского расклада несколькими способами. Мы не стремимся организовать всецело вероятностную модель текста. Мы используем оптимизацию взамен предшествующих распределений. Мы предпочитаем использовать алгоритм максимизации математического ожидания (EM) вместо сложного вывода по методу Байеса из-за его простоты в реализации, эффективности и легкости интерпретации результатов. EM-алгоритм обладает прямой итеративной процедурой оптимизации, что облегчает его применение на практике и упрощает процесс анализа данных. Этот метод также может быть более эффективным в вычислительном отношении, что особенно важно при работе с большими объемами данных.

Алгоритм ARTM, использующий EM-алгоритм, делает процесс планирования и вывода модели тематического моделирования более простым и доступным для исследователей и практиков, что облегчает построение и анализ тематических моделей.

Распределения слов и тем, которые мы приобретаем через ARTM, могут быть легко разреженными. Это может существовать нежелательно. Мы можем употреблять регуляризацию, для уменьшения этой разреженности и сделать модель наиболее устойчивой. Ключевая мысль ARTM - прибавить добавочную информацию о цели в нашу оптимизируемую функцию (4).

$$R(\Phi,\Theta) = \sum_i \tau_i R_i(\Phi,\Theta)\,, L(\Phi,\Theta) + R(\Phi,\Theta) \rightarrow^{max}_{\Phi,\Theta,}. \qquad (4)$$

Коэффициенты регуляризации, обозначаемые как $\tau \geq 0$, настраиваются экспериментально и контролируют воздействие регуляризатора на процесс обучения модели тем. В обучении модели тем используется алгоритм

максимизации правдоподобия (EM).

Метод ARTM-EM представляет собой алгоритм, включающий в себя этот процесс. На входе алгоритма указывается сумма документов (D) и количество тем (|T|), а также исходные приближения матриц Θ и Φ. Этот алгоритм обеспечивает эффективное обучение тематической модели с использованием EM-подхода и позволяет управлять регуляризацией для достижения желаемых результатов (рисунок 4);

Шаг 1: Инициализировать столбцы вектора ϕ_t;

Шаг 2: Случайным образом задать ϕ_t и θ_d;

Шаг 3: Обнулить $\hat{n}_{wt}$ и $\hat{n}_{dt}$ для всех $d \in D, w \in W, t \in T$;

Шаг 4: Для всех $d \in D, w \in d$;

Шаг 5: Вычислить $p(w|d) = \sum_{t \in T} \phi_{wt}\theta_{td}$;

Шаг 6: Для всех $t \in T$;

Шаг 7: Вычислить $p(t|d,w) = \frac{\phi_{wt}\theta_{td}}{p(w|d)}$;

Шаг 8: Увеличить $\hat{n}_{wt}$ и $\hat{n}_{td}$ на $n_{dw} \cdot p(t|d,w)$;

Шаг 9: Обновить $\phi_{wt} \propto (\hat{n}_{wt} + \phi_{wt} \cdot \frac{\partial R}{\partial \phi_{wt}})_+$ для всех $w \in W, t \in T$;

Шаг 10: Обновить $\theta_{td} \propto (\hat{n}_{td} + \theta_{td} \cdot \frac{\partial R}{\partial \theta_{td}})_+$ для всех $d \in D, t \in T$;

Шаг 11: Повторять до тех пор, пока Θ и Φ не сойдутся.

Рисунок 4 – Метод ARTM-максимизации правдоподобия EM

Коэффициенты регуляризации необходимо выбирать вручную, поскольку они контролируют влияние регуляризаторов на процесс обучения модели и могут существенно влиять на качество и интерпретируемость результатов. В ARTM используются различные типы регуляризаторов, такие как сглаживание, разреживание и декорреляция, каждый из которых оказывает

свое воздействие на модель.

Например, стандартизация разреженности тем может привести к тому, что модель будет искать очень мало тем для представления других документов, что может затруднить извлечение интерпретируемых тем. Поэтому выбор коэффициентов регуляризации является важным шагом при построении модели, и его следует выполнять внимательно, исходя из конкретных целей и требований к модели.

Разреживающий регулязатор параметров модели (5):

$$R(\Phi,\Theta) = -\beta_0 \sum_{t\in T}\sum_{w\in W}\beta_w \ln \varphi_{wt} - \alpha_0 \sum_{d\in D}\sum_{t\in T}\alpha_t \ln \theta_{td} \rightarrow^{max}_{\Phi,\Theta}. \quad (5)$$

Сглаживание распределений терминов в темах. Применяется ради выделения низкоприоритетных тем, собирающих совместную лексику языка либо всеобщую лексику предоставленной коллекции.

Сглаживающий регуляризатор (6):

$$R(\Phi,\Theta) = \beta_0 \sum_{t\in T}\sum_{w\in W}\beta_w \ln \varphi_{wt} + \alpha_0 \sum_{d\in D}\sum_{t\in T}\alpha_t \ln \theta_{td} \rightarrow^{max}_{\Phi,\Theta}. \quad (6)$$

Декоррелирующий регуляризатор R(Ф) пытается уменьшить связь между темами, минимизируя ковариации между столбцами в матрице Ф. Здесь Ф\PhiФ представляет собой матрицу вероятностей слов для каждой темы t из множества тем T, а W – множество слов (7):

$$R(\Phi) = -\tau \sum_{t\in T}\sum_{s\in \frac{T}{t}}\sum_{w\in W}\varphi_{wt}\,\varphi_{ws} \rightarrow^{max}_{\Phi}, \quad (7)$$

где $\tau \geq 0$ – регуляризационный коэффициент.

Декоррелирующий регуляризатор (7) стремится сделать распределения слов в различных темах преимущественно различными. Это означает, что он

пытается уменьшить схожесть слов, которые встречаются в разных темах, делая их распределения более независимыми друг от друга. Конкретно, это означает, что слова, которые встречаются часто в одной теме, будут иметь меньшую вероятность встречаться в других темах, что способствует созданию более разнообразных и неповторяющихся тематических профилей.

Он делает это, прибавляя штраф из-за подобия среди тем к функции утрат модели. Иногда слова в разных темах меньше схожи друг с другом, любая тема делается больше непохожей от других.

Различные регуляризаторы в моделях тематического моделирования направлены на увеличение уникальности тем. Например, декоррелирующий регуляризатор старается сделать распределения слов в различных темах преимущественно различными, что способствует созданию более неповторяющихся тем. Регуляризаторы, такие как разреживающие и сглаживающие, контролируют разреженность и сглаженность распределений слов в темах, что также способствует уникальности тем.

Разделение документов на предложения и организация иерархических моделей позволяют более точно выявлять тематическую структуру текста. Это улучшает интерпретируемость и точность модели, так как позволяет учитывать более мелкие детали и взаимосвязи между словами в тексте.

Метод ARTM позволяет соединять различные регуляризаторы в единую модель благодаря своей гибкости и механизму обновления параметров (M-шаг). M-шаг алгоритма ARTM позволяет оптимизировать модель, учитывая различные регуляризаторы, что делает возможным создание сложных моделей с учетом различных аспектов текста и задач анализа. Такой подход позволяет получать более точные и интерпретируемые результаты тематического моделирования.

Для анализа и сравнения алгоритмов тематического моделирования в контексте новостей мы можем обратиться к таблице 4.

Таблица 4 – Сравнение алгоритмов тематического моделирования в новостях

Метод ТМ	Алгоритм	Преимущества	Недостатки
PLSA	Использует способность вычислительных моделей оценивать вероятности для определения скрытых тем в текстовых данных.	Способность точно выявлять скрытые темы в тексте, что помогает лучше понять содержание документов и организовать их для анализа.	Не учитывает очередь слов в тексте и не способен имитировать взаимоотношение среди тем.
LDA	Применяет алгоритм Гиббса с дополнительным расчетом перплексии для определения параметров модели.	Презентует любой текст будто вероятностное расположение тем и предусматривает вероятности текстов в любой теме.	Сможет организовывать ненужные темы или игнорировать значительные ньюансы документов.
ARTM	Применяет регуляризаторы.	Кроме нахождения тем, ARTM может обдумывать иные характеристики документов и подключать наружные факторы.	Просит усиленной настройки параметров модификации ради извлечения правильных результатов.

Анализ показывает, что выбор подходящего алгоритма должен обеспечивать и точность моделирования, и понятность результатов.

2.5 Выбор тематической модели анализа новостей

С увеличением численности и многообразия цифровой информации становится все сложнее замечать необходимую информацию. Однако, с поддержкой разнообразных способов разбора текстов, подобных вероятностным тематическим моделям, возможно упростить отбор и познание информации в больших объёмах данных. Есть многочисленные инструменты, которые помогают исследовать и визуализировать текстовые данные, сооружая данный ход наиболее лёгким и ясным для пользователей. Исследование и ссылки – это два самых распространенных инструмента для работы с онлайн-информацией. Поиск помогает обнаруживать информацию

по ключевым словам либо текстам, внедренным пользователем. Ссылки разрешают заходить на остальные сайты и ресурсы, сопряженные с интересами пользователя. Это хорошее средство взаимодействовать с нашим онлайн-архивом, но данного недостаточно.

Вероятностные тематические модели ориентируют доставить любой документ будто комплект тем с разными вероятностями, а любую тему – будто набор слов с разными вероятностями. Вероятностные тематические модели ориентированы на выявление скрытых тематик в текстовых данных, не ограничиваясь только ключевыми словами. Они анализируют совокупность слов в документе и пытаются выделить скрытые темы или концепции, которые объясняют смысл текста. Таким образом, вместо того чтобы просто искать документы по ключевым словам, эти модели позволяют выявить более глубокие и комплексные тематические структуры в тексте.

Вероятностная тематическая модификация, включающая коллекцию текстовых документов (D), словарь терминов (W) и конечное множество тем (T), представляет собой задачу анализа текстовых данных с использованием методов тематического моделирования. Основная цель состоит в применении таких моделей, которые позволят извлечь скрытые тематические структуры из документов, представленных в коллекции D, с учетом словаря терминов W и заданного числа тем из множества T.

Для успешного выполнения задачи необходимо определить оптимальные параметры модели, такие как число тем и использование регуляризаторов, которые могут улучшить качество моделирования. Оптимальное количество тем обычно выбирается экспериментально и зависит от специфики данных и целей исследования.

Полученные темы могут быть интерпретированы и использованы для категоризации и анализа текстовых документов в коллекции. Они позволяют выделить ключевые тематики и основные направления в текстах, что облегчает понимание содержания и организацию данных для дальнейшего исследования.

Вероятностная модель для создания данных обобщает анализ условной вероятности, формулу полной вероятности и гипотезу условной независимости [4]. Эти концепции позволяют моделировать взаимосвязи и вероятностные закономерности в данных. Условная вероятность рассматривает вероятность события при условии наступления другого события, формула полной вероятности учитывает все возможные исходы события, а гипотеза условной независимости предполагает, что некоторые события могут рассматриваться как независимые в определенных условиях. Эти концепции объединяются в вероятностной модели для анализа и моделирования данных в различных областях, обеспечивая более глубокое понимание их структуры и взаимосвязей (8):

$$p(w|d) = \sum_{t \in T} p(w|t)p(t|d), \quad (8)$$

где p(t|d) и p(w|t) – вероятности распределения слов.

Согласно модели генерации данных, формула описывает вероятность появления слова w в документе d как сумму произведений вероятностей появления слова в теме и вероятностей темы в документе. Здесь p(t|d) и p(w|t) представляют собой искомые распределения.

Алгоритм генерации коллекции текстов с использованием вероятностной модели: распределений p(t|d)p(t|d)p(t|d) и p(w|t)p(w|t)p(w|t), набор пар ((d_i, w_i)), (i = 1,…, n); (рисунок 5):

Шаг 1: Для каждого документа d в коллекции D.

Шаг 2: Определение длины n_d для документа d.

Шаг 3: Для каждого $i = 1, \ldots, n_d$:

Шаг 4: Случайным образом выбирается тема t из распределения $p(t|d)$.

Шаг 5: Выбирается независимый термин w из распределения $p(w|t)$.

Шаг 6: Добавляется пара (d, w) в набор, при этом тема t "забывается".

Рисунок 5 – Алгоритм генерации коллекции текстов

Процесс генерации последовательности текстов документа показан на рисунке 6.

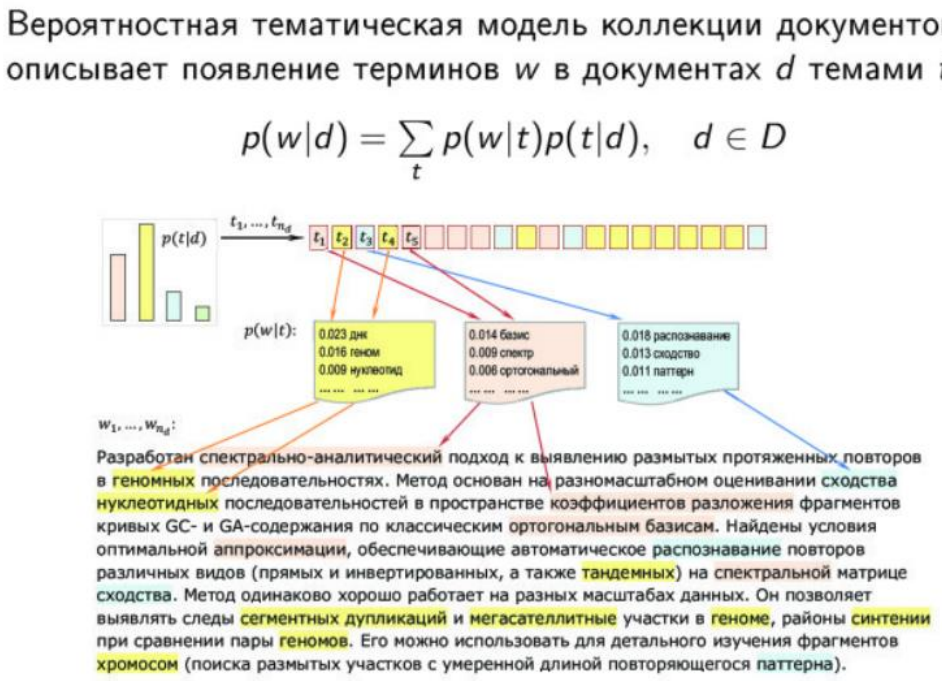

Рисунок 6 – Процесс функционирования вероятностной модели

Процесс генерации последовательности текстовых документов с учетом

распределений (p(t|d)) и (p(w|t)) включает выбор темы для каждого документа из соответствующего распределения (p(t|d)) и последующий выбор слов из распределения (p(w|t)), определяющего слова, характерные для выбранной темы. Этот процесс повторяется для каждого документа в коллекции, обеспечивая генерацию текстовых документов с уникальными тематическими направлениями и содержанием.

Произвольный документ в коллекции организуется с учетом его распределения по темам. Для каждого документа определяется вероятностное распределение по темам, то есть определяется, насколько каждая тема представлена в данном документе. Затем, для каждой темы, из распределения слов для этой темы выбираются слова, соответствующие содержанию этой темы. Таким образом, организация документа включает в себя учёт тематической структуры документа и выбор слов, которые наиболее соответствуют этим темам. Это позволяет представить документ в виде комбинации тем и соответствующих им слов, что обеспечивает его организацию и структурирование в коллекции.

В работе [15] рассмотрен известный и применяемый в реальных условиях метод LDA. Исследователи применяют разнообразные тематические модели в своей работе, выбирая методы в зависимости от целей и характеристик исследования. Некоторые из наиболее популярных методов включают максимальное правдоподобие (MLE), сингулярное разложение (SVD), метод моментов, алгоритм на основе неотрицательной матричной факторизации (NMF), вероятностные тематические модели, вероятный латентно-семантический анализ (LSA) и латентное размещение Дирихле (LDA).

Выводы по главе 2

Глава 2 содержит обзор и анализ алгоритмов тематического моделирования. Исследование этих алгоритмов привело к следующим выводам:

- LDA применяет вероятностную модель для объяснения формирования текстовых документов на основе скрытых тем и вероятностей слов в каждой теме;
- PLSA также основана на вероятностной модели, но учитывает только распределения слов в документах, что иногда приводит к неточностям из-за игнорирования связей между словами;
- в отличие от LDA и PLSA, ARTM может применять как вероятностные, так и эвристические методы для выделения тем;
- ARTM стремится улучшить качество моделирования тем посредством введения регуляризации, учитывающей предположения о структуре документов. Это позволяет, например, уменьшать пересечение тем или повышать интерпретируемость результатов;
- алгоритмы тематического моделирования играют ключевую роль в анализе текстовых данных, помогая выявлять скрытые темы и связи в больших объемах текстов;
- сравнительный анализ показывает, что LDA и PLSA имеют ограничения в точности моделирования, особенно при работе с короткими или слабосвязанными документами. ARTM является более гибким и точным методом, способным учитывать все факторы модели, что делает его более продвинутым подходом к тематическому моделированию.

Глава 3 Реализация и тестирование ПО

3.1 Реализация ПО

Можно употребить несколько библиотек, для того, чтобы скопить и проанализировать материал о том, как разрабатывается программное обеспечение для сбора новостей и оценки их активности.

Существует подобная нужная библиотека для Python под названием Beautiful Soup. Она помогает вам получать сведения с веб-страниц, устраняя из них лишние элементы. Она особенно отлично годится для анализа HTML и XML документов. Кроме того, с помощью Beautiful Soup вы можете свободно замечать необходимые вам доли страницы и изменять их.

Еще одна нужная библиотека – Pandas. Она представляет собой инструментарий для обработки информации в Python. С ее поддержкой возможно брать и исследовать информацию из разных источников, включительно веб-сайты и API. Pandas разрешает разбирать материал в различных форматах, подобных CSV, Excel, SQL, а также производить их очистку и преображение [15].

Для работы с текстами и анализа их смысла зачастую применяют Natural Language Toolkit (NLTK). С его поддержкой возможно делать различные задачи, такие как разбиение текста на единичные слова или предложения, отбор ключевых слов и установление долей речи. Также в NLTK имеются готовые модели для определения расположения текста, определения именованных сутей и прочих вопросов обрабатывания текста.

Для машинного обучения весьма полезна библиотека scikit-learn. В ней множество различных методов, которые ориентируют производить классификацию, кластеризацию и прогнозирование. Эти методы ориентируют нам осмыслить закономерности в данных и делать мониторинги о грядущих тенденциях.

В общем, мы можем перемешивать и сочетать данные библиотеки, чтобы разработать программное обеспечение, которое будет собирать новости и соединять наиболее популярные тенденции. На листинге 1 (рисунок 7) представлен код программы.

```
import re
from gensim.models.doc2vec import Doc2Vec, TaggedDocument
from gensim.corpora.dictionary import Dictionary
from gensim.models import LdaModel
import pyLDAvis.gensim_models
from sklearn.metrics.pairwise import cosine_similarity

# URL для сбора новостей
url = 'https://dzen.ru/news'
news_articles = fetch_news(url)

# Тематическое моделирование с использованием LDA
dictionary = Dictionary(preprocessed_news)
corpus = [dictionary.doc2bow(text) for text in preprocessed_news]

lda_model = LdaModel(corpus, num_topics=10, id2word=dictionary, passes=15)

topics = lda_model.print_topics(num_words=4)
for topic in topics:
 print(topic)

# Визуализация тем
pyLDAvis.enable_notebook()
vis = pyLDAvis.gensim_models.prepare(lda_model, corpus, dictionary)
pyLDAvis.display(vis)

def find_similar_documents(doc_id, document_vectors, top_n=5):
 target_vector = document_vectors[doc_id]
 similarities = []
 for i, vector in enumerate(document_vectors):
 if i != doc_id:
 similarity = compute_similarity(target_vector, vector)
 similarities.append((i, similarity))
 similarities = sorted(similarities, key=lambda x: x[1], reverse=True)
 return similarities[:top_n]

similar_docs = find_similar_documents(0, document_vectors)
print("Наиболее похожие документы на первый документ:")
for doc_id, similarity in similar_docs:
 print(f"Документ {doc_id} с сходством {similarity}")
```

Рисунок 7 – Листинг ПО

Функция предобработки текста показана 8 рисунке.

```
def preprocess_text(text):
    text = text.lower()
    text = re.sub(r'\d+', '', text)
    text = re.sub(r'[^\w\s]', '', text)
    tokens = word_tokenize(text)
    stop_words = set(stopwords.words('english'))
    tokens = [word for word in tokens if word not in stop_words]
    lemmatizer = WordNetLemmatizer()
    tokens = [lemmatizer.lemmatize(word) for word in tokens]
    return tokens

preprocessed_news = [preprocess_text(article) for article in news_articles]
```

Рисунок 8 – Функция предобработки текста

Рассмотрим процесс тестирования ПО.

3.2 Тестирование программного обеспечения

При тесте приложения был использован подход, который именуется функциональным тестированием.

Функциональное тестирование - это проверка программы или системы на то, чтобы убедиться, что она работает так, как задумано, и соответствует требованиям, которые были установлены для нее. Этот метод включает в себя тестирование различных функций программы, чтобы удостовериться, что они работают правильно и удовлетворяют нужды пользователей [6].

Цель функционального тестирования заключается в проверке того, что программа делает все, что от нее ожидается, и не делает ничего лишнего.

Пользовательский интерфейс (UI) – это часть программы или веб-сайта, с которой пользователи взаимодействуют. Он включает в себя дизайн и функциональность кнопок, меню, форм и других элементов. Хороший пользовательский интерфейс легок в использовании, понятен и приятен внешне, что делает пользователей более довольными и заинтересованными.

Наверху экрана находится контрольная панель с тремя кнопками: «Ваша подборка», «Все новости» и «Профиль» (рисунок 9).

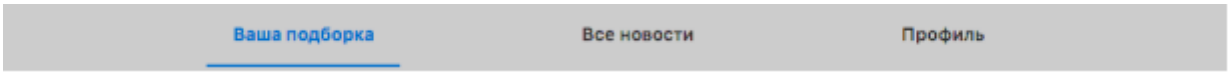

Рисунок 9 – Управление навигацией

Первая вкладка, «Ваша подборка» (рисунок 10), демонстрирует персонализированные новости, базированные в предпочтениях пользователя. Когда читатель не вошёл в профиль, эта вкладка отображаться не будет.

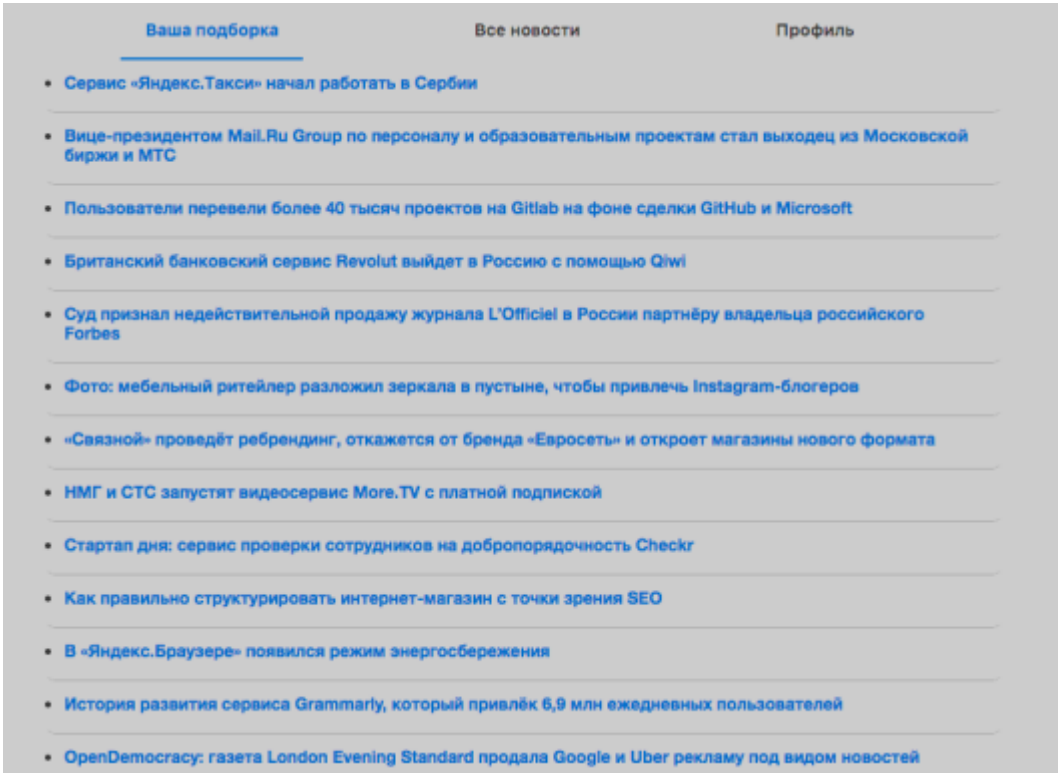

Рисунок 10 – Вкладка «Ваша подборка»

Заголовок «Все новости» доступна всем пользователям, даже если они не регистрировались в профиль. Тут представляются все новости без разделения по темам (рисунок 11).

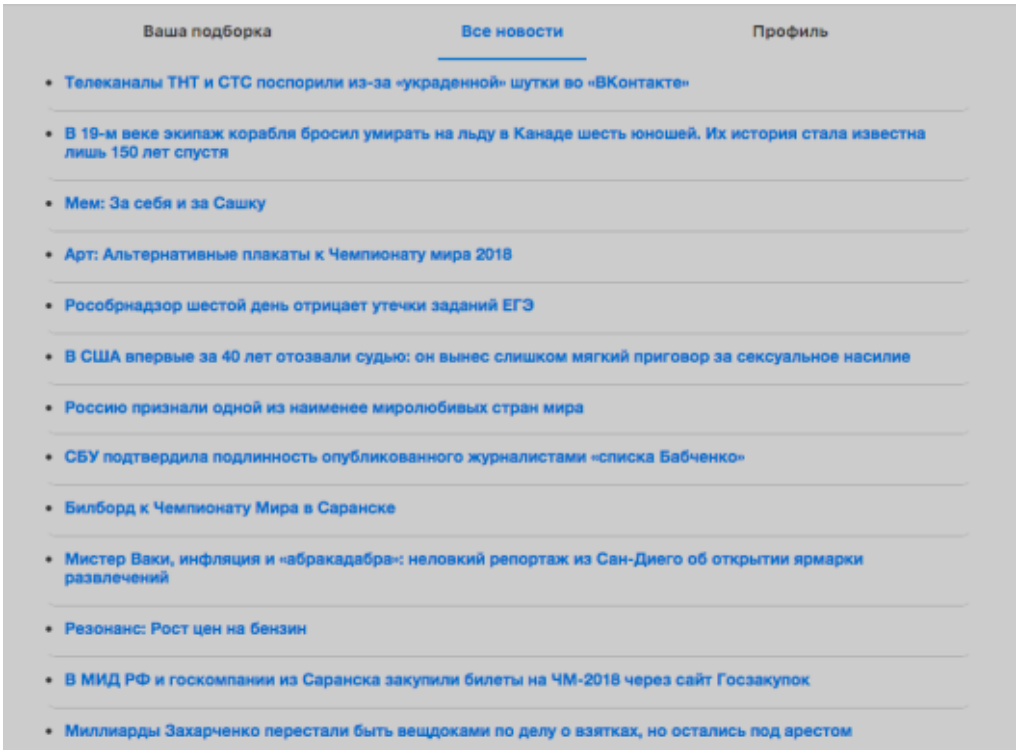

Рисунок 11 – Заголовок "Все новости"

Вкладка «Профиль» предназначена для регистрации и входа в систему. При первом посещении демонстрируется страница для входа и клавиша для регистрации (рисунок 12).

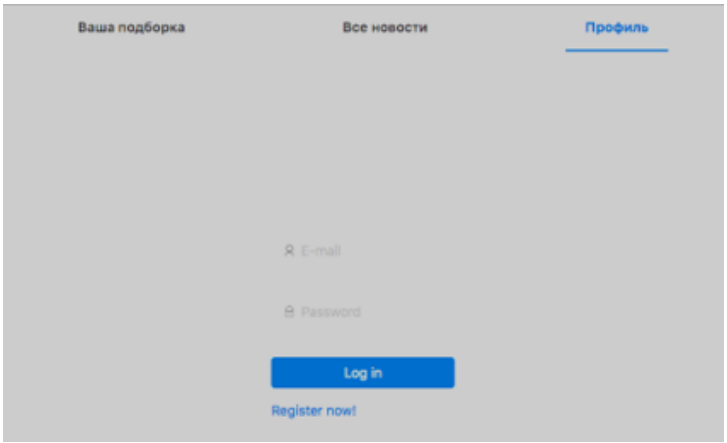

Рисунок 12 – Страница «Профиль»

Функция коррекции пунктуации и стоп-слов показана на рисунке 13.

```
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')

def preprocess_text(text):
    # Приведение к нижнему регистру
    text = text.lower()

    # Удаление пунктуации и чисел
    text = re.sub(r'\d+', '', text)
    text = re.sub(r'[^\w\s]', '', text)

    # Токенизация
    tokens = word_tokenize(text)

    # Удаление стоп-слов
    stop_words = set(stopwords.words('english'))
    tokens = [word for word in tokens if word not in stop_words]
```

Рисунок 13 – Коррекция стоп-слов и пунктуация

Первоначально мы переделаем тексты в наборы численностей с поддержкой способа Doc2Vec. Данный способ возможно представлять полной версией Word2Vec, который создаёт числовое понятие ради любого документа. Оба метода, Word2Vec и Doc2Vec, находятся в библиотеке Python под наименованием Gensim.

Word2Vec применяет нейронные сети, для присвоения любому слову числовое представление. Он обрабатывает огромные размеры текста и автоматом разыскивает связи среди слов. Таким образом, любое слово в тексте приобретает свой комплект чисел, отображающих связи с другими словами (рисунок 14).

```
from gensim.models.doc2vec import Doc2Vec, TaggedDocument

tagged_data = [TaggedDocument(words=_d, tags=[str(i)]) for i, _d in enumerate(preprocessed_news)]

model = Doc2Vec(vector_size=20, alpha=0.025, min_alpha=0.00025, min_count=1, dm=1)
model.build_vocab(tagged_data)

for epoch in range(100):
    print(f'Iteration {epoch}')
    model.train(tagged_data, total_examples=model.corpus_count, epochs=model.epochs)
    model.alpha -= 0.0002
    model.min_alpha = model.alpha

model.save("d2v.model")
```

Рисунок 14 – Векторизация слов и презентация в виде словаря

Файлы читаются отдельными документами каждый раз, а не вместе сразу. Это позволяет действовать с весьма огромными коллекциями документов, которые не вмещаются всецело в память компьютера.

Для понимания, как документы в нашей коллекции схожи друг с другом, мы используем расположение тем в отдельном документе.

На рисунке 15 представлен код, который реализует способ для определения приближения среди документов.

```
# Загрузка обученной модели Doc2Vec
model = Doc2Vec.load("d2v.model")

# Преобразование документов в векторы
document_vectors = [model.infer_vector(doc) for doc in preprocessed_news]

# Функция для вычисления косинусного сходства
def compute_similarity(vector1, vector2):
    return cosine_similarity([vector1], [vector2])[0][0]

# Пример: нахождение наиболее близких документов к первому документу
def find_similar_documents(doc_id, document_vectors, top_n=5):
    target_vector = document_vectors[doc_id]
    similarities = []

    for i, vector in enumerate(document_vectors):
        if i != doc_id:
            similarity = compute_similarity(target_vector, vector)
            similarities.append((i, similarity))

    # Сортировка по сходству и выбор top_n
    similarities = sorted(similarities, key=lambda x: x[1], reverse=True)
    return similarities[:top_n]

# Найти 5 наиболее похожих документов на первый документ
similar_docs = find_similar_documents(0, document_vectors)
print("Наиболее похожие документы на первый документ:")
for doc_id, similarity in similar_docs:
    print(f"Документ {doc_id} с сходством {similarity}")
```

Рисунок 15 – Реализация близости документов с помощью обученной модели

Для того, чтобы исключить запутанности из-за крупного численности информации, мы постановили избрать лишь 20 беспорядочных документов для отображения на графике. Если бы мы использовали все 3000 документов, диаграмма была бы чрезмерно перегруженной. Была создана матрица близости текстов (рисунок 16).

```
[[1.00, 0.82, 0.74, 0.56, 0.67, 0.49, 0.71, 0.62, 0.85, 0.77, 0.66, 0.88, 0.64, 0.78, 0.59, 0.70, 0.81, 0.53, 0.79, 0.69],
 [0.82, 1.00, 0.68, 0.52, 0.60, 0.43, 0.66, 0.58, 0.80, 0.73, 0.61, 0.83, 0.60, 0.75, 0.56, 0.64, 0.77, 0.51, 0.76, 0.63],
 [0.74, 0.68, 1.00, 0.47, 0.55, 0.39, 0.58, 0.51, 0.70, 0.65, 0.55, 0.73, 0.53, 0.68, 0.50, 0.57, 0.67, 0.45, 0.70, 0.58],
 [0.56, 0.52, 0.47, 1.00, 0.42, 0.31, 0.46, 0.40, 0.57, 0.54, 0.44, 0.60, 0.43, 0.56, 0.42, 0.48, 0.54, 0.37, 0.55, 0.45],
 [0.67, 0.60, 0.55, 0.42, 1.00, 0.38, 0.55, 0.48, 0.68, 0.62, 0.52, 0.71, 0.50, 0.64, 0.48, 0.55, 0.63, 0.43, 0.65, 0.53],
 [0.49, 0.43, 0.39, 0.31, 0.38, 1.00, 0.41, 0.35, 0.50, 0.46, 0.39, 0.52, 0.37, 0.47, 0.36, 0.41, 0.48, 0.33, 0.49, 0.40],
 [0.71, 0.66, 0.58, 0.46, 0.55, 0.41, 1.00, 0.53, 0.72, 0.65, 0.54, 0.76, 0.52, 0.68, 0.51, 0.57, 0.66, 0.43, 0.70, 0.57],
 [0.62, 0.58, 0.51, 0.40, 0.48, 0.35, 0.53, 1.00, 0.63, 0.58, 0.49, 0.66, 0.45, 0.58, 0.44, 0.50, 0.58, 0.38, 0.60, 0.49],
 [0.85, 0.80, 0.70, 0.57, 0.68, 0.50, 0.72, 0.63, 1.00, 0.78, 0.67, 0.81, 0.61, 0.75, 0.59, 0.69, 0.80, 0.51, 0.77, 0.67],
 [0.77, 0.73, 0.65, 0.54, 0.62, 0.46, 0.65, 0.58, 0.78, 1.00, 0.60, 0.74, 0.56, 0.69, 0.52, 0.61, 0.70, 0.48, 0.72, 0.61],
 [0.66, 0.61, 0.55, 0.44, 0.52, 0.39, 0.54, 0.49, 0.67, 0.60, 1.00, 0.67, 0.53, 0.62, 0.48, 0.54, 0.61, 0.44, 0.64, 0.53],
 [0.88, 0.83, 0.73, 0.60, 0.71, 0.52, 0.76, 0.66, 0.81, 0.74, 0.67, 1.00, 0.63, 0.79, 0.61, 0.70, 0.80, 0.53, 0.78, 0.69],
 [0.64, 0.60, 0.53, 0.43, 0.50, 0.37, 0.52, 0.45, 0.61, 0.56, 0.53, 0.63, 1.00, 0.60, 0.47, 0.52, 0.60, 0.42, 0.62, 0.50],
 [0.78, 0.75, 0.68, 0.56, 0.64, 0.47, 0.68, 0.58, 0.75, 0.69, 0.62, 0.79, 0.60, 1.00, 0.56, 0.64, 0.72, 0.48, 0.73, 0.61],
 [0.59, 0.56, 0.50, 0.42, 0.48, 0.36, 0.51, 0.44, 0.59, 0.52, 0.48, 0.61, 0.47, 0.56, 1.00, 0.50, 0.57, 0.40, 0.58, 0.47],
 [0.70, 0.64, 0.57, 0.48, 0.55, 0.41, 0.57, 0.50, 0.69, 0.61, 0.54, 0.70, 0.52, 0.64, 0.50, 1.00, 0.64, 0.46, 0.65, 0.54],
 [0.81, 0.77, 0.67, 0.54, 0.63, 0.48, 0.66, 0.58, 0.80, 0.70, 0.61, 0.80, 0.60, 0.72, 0.57, 0.64, 1.00, 0.51, 0.73, 0.63],
 [0.53, 0.51, 0.45, 0.37, 0.43, 0.33, 0.43, 0.38, 0.51, 0.48, 0.44, 0.53, 0.42, 0.48, 0.40, 0.46, 0.51, 1.00, 0.52, 0.44],
 [0.79, 0.76, 0.70, 0.55, 0.65, 0.49, 0.70, 0.60, 0.77, 0.72, 0.64, 0.78, 0.62, 0.73, 0.58, 0.65, 0.73, 0.52, 1.00, 0.66],
 [0.69, 0.63, 0.58, 0.45, 0.53, 0.40, 0.57, 0.49, 0.67, 0.61, 0.53, 0.69, 0.50, 0.61, 0.47, 0.54, 0.63, 0.44, 0.66, 1.00]]
```

Рисунок 16 – Матрица близости текстов

Модель LDA доступна в библиотеке Gensim. Её основное превосходство в том, что её можно дообучать. В отличие от моделей LSA и pLSA, которые производят обучение с начала при добавлении свежего документа, в LDA возможно элементарно прибавить свежие материалы и дообучить модель. Образец данного процесса представлен на рисунке 17.

```
# Обучение модели LDA
lda_model = LdaModel(corpus, num_topics=10, id2word=dictionary, passes=15)
```

Рисунок 17 – Обучение тематической модели LDA

Для отображения информации мы применяем инструментарий под

названием LDAVis. С помощью библиотеки pyLDAvis модель LDA переустроена способом PCA (анализ главных компонент), для снижения размерности до двух измерений. Это делается для лучшего изображения предоставленных на графике. На графике (рисунок 18) видимы круги, каждый из них презентует одну из сокрытых тем в пространстве. Дистанция среди кругов показывает, как темы схожи либо различны.

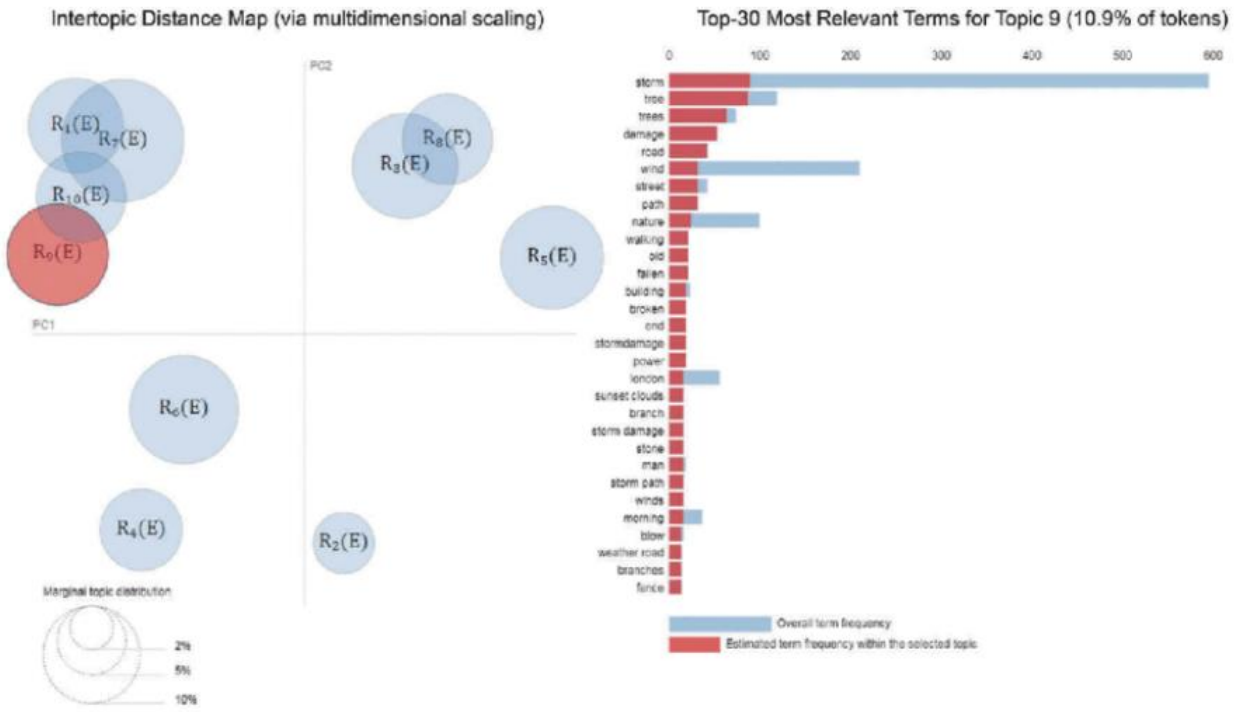

Рисунок 18 – Функциональный график pyLDAvis

Диаграмма LDAvis раздроблена на две части: в первой части представлено простое двумерное изображение информации LDA, которая сначала обладала множеством измерений.

Второй элемент - это изменяемая гистограмма, изменяющаяся в связи от операций пользователя. Обе части презентованы на рисунке 18.

В правой гистограмме слова показываются по убыванию их важности. Подъем столбиков демонстрирует частоту встречаемости слов. Красный цвет показывает на частоту слова внутри предоставленной темы, а синий - на

общую частоту слова во всей коллекции документов.

Для визуализации информации с обилием измерений мы используем способ понижения размерности.

Например, когда у нас имеются сведения с тысячами параметров, сокращение размерности ориентирует адаптировать их исследование и визуализацию.

Способ t-SNE описывает соответствие среди точек на основе расстояний: ближние точки числятся похожими, а далекие - непохожими. Мы выбираем 500 самых частых слов и отображаем их на графике (рисунки 19 и 20).

```
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

# Преобразование данных с использованием PCA
pca = PCA(n_components=2)
pca_result = pca.fit_transform(data)  # data - ваш набор данных

# Визуализация результата снижения размерности
plt.scatter(pca_result[:, 0], pca_result[:, 1], alpha=0.5)
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('PCA Visualization')
plt.show()
```

Рисунок 19 – Код для снижения размерности информации

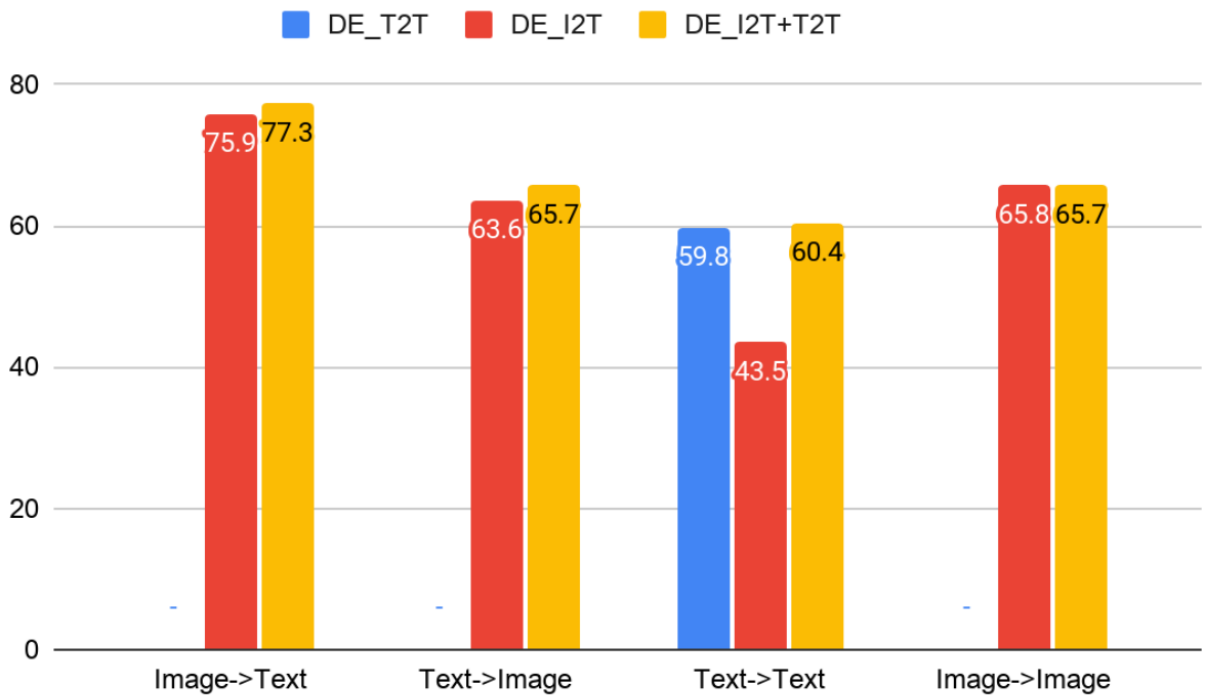

Рисунок 20 – График сходства текстов

Способ t-SNE ориентирует нам обнаруживать различные методы представления одной и той же цифры либо замечать похожие слова либо фразы с подобным значением при анализе текстов.

Отчего нам не использовать исключительно LDA? Дело в том, что для тематического моделирования есть множество разных подходов, и для каждой определенной темы разбора текстов возможно понадобится применение иных методов. При выборе подобающего способа существенно считаться с установкой исследования, а также специфики данных.

BigARTM - это мощный инструментарий для тематического моделирования, который дозволяет анализировать огромные размеры текстовой информации и использовать разные способы и регуляризаторы ради усовершенствования качества модели. Он может вычислять разные задачи, подобные разделению, сглаживанию, декорреляция тем и другим. Композиция регуляризаторов в BigARTM возможно значительно повысит немного характеристик свойств модели, не усложняя её структуру.

BigARTM предоставляет инструменты для настройки и улучшения тематических моделей, включая регуляризацию, тюнинг гиперпараметров, итеративное улучшение модели, оценку качества, интерактивную визуализацию и интеграцию с другими инструментами. Эти возможности позволяют исследователям и практикам эффективно управлять процессом создания и улучшения тематических моделей с помощью BigARTM.

Регуляризаторы SmoothSparsePhi, SmoothSparseTheta и DecorrelationPhi применяются для контроля разреженности и декоррелированности тематических моделей в BigARTM, что способствует их уникальности и интерпретируемости. Импортирует класс ARTM из модуля BigARTM. После этого вы сможете использовать BigARTM для создания и обучения тематических моделей, а также для применения различных регуляризаторов и других методов анализа текстовых данных (рисунок 21).

```
from sklearn.manifold import TSNE
import matplotlib.pyplot as plt

# Преобразование данных с использованием t-SNE
tsne = TSNE(n_components=2, perplexity=30.0)
tsne_result = tsne.fit_transform(data)  # data - ваш набор данных

# Визуализация результата представления данных в виде одного класса
plt.scatter(tsne_result[:, 0], tsne_result[:, 1], alpha=0.5)
plt.xlabel('t-SNE Component 1')
plt.ylabel('t-SNE Component 2')
plt.title('t-SNE Visualization')
plt.show()
```

Рисунок 21 – визуализируем информацию в виде единого класса

API Python в библиотеке действует приблизительно так же, как алгоритмы из scikit-learn. Он принимает входные сведения в виде особого класса, именуемый BatchVectorizer, как изображено на рисунке 21.

Словарь (Dictionary) - это особый инструмент в BigARTM, который

включает информацию о коллекции текстов (например, слова, счетчики и остальные сопряженные значения). Мы используем данный словарь для создания матрицы Ф, как показано на рисунке 22.

```
# Создание словаря и корпуса
dictionary = Dictionary(preprocessed_news)
corpus = [dictionary.doc2bow(text) for text in preprocessed_news]
```

Рисунок 22 – Разработка словаря с информацией о собрании текстов

ARTM разрешает пользоваться всеми способностями BigARTM и сохранять значения при любом обновлении матрицы Ф. Вдобавок мы можем исследовать личные способы регуляризации ради наилучшей установки модификации под ваши нужды. Аддитивная стандартизация разрешает объединять разные способы регуляризации ради усовершенствования свойств модели, учитывая различные аспекты одновременно, например понятность тем, их разновидность и разреженность матрицы слов-тем. ARTM вдобавок удерживает относительную регуляризацию, что позволяет использовать способы регуляризации исключительно к назначенным группам тем или документов.

Анализ свойства тематических моделей сложна, поэтому мы не знаем верных тем любого документа. Впрочем, мы можем использовать перплексию, оценивающая, как именно модификация справляется с испытательной выборкой. Наименьшее значение перплексии обозначает наилучшую модель.

Хоть и перплексия нужна для оценки качества модели, она не является единой метрикой в тематическом моделировании. Существенно еще и представление приобретенных тем, их уместность и согласованность контексту.

После сравнения моделей LDA, PLSA и ARTM, на рисунке 23 изображена поправка перплексии в процессе обучения данных модификаций.

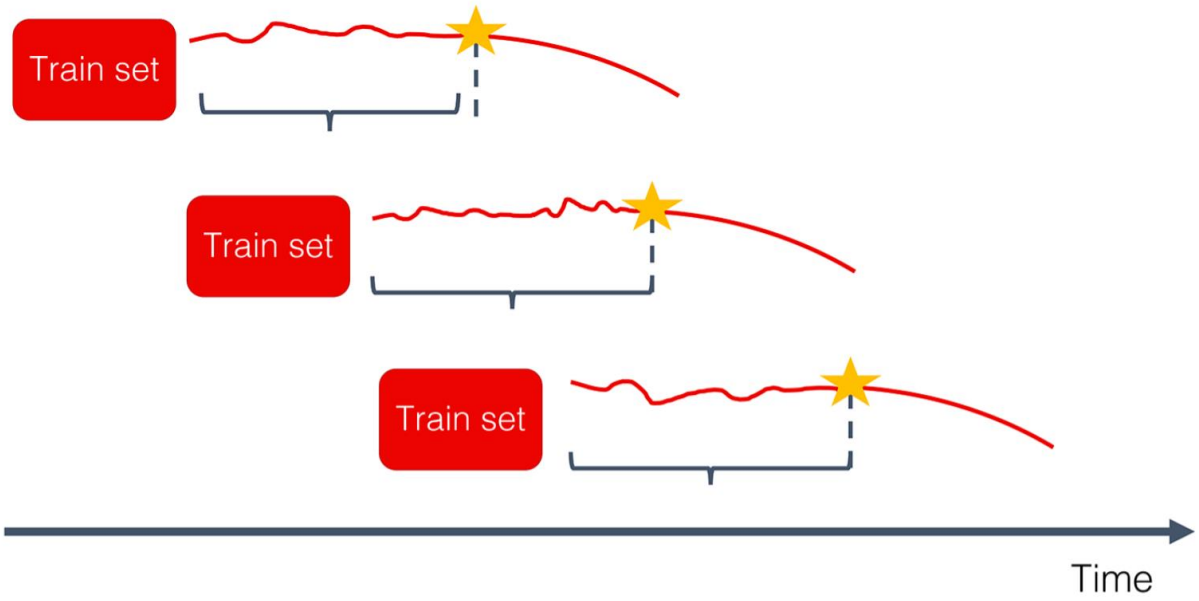

Рисунок 23 – График изменения переплексии во время обучения

Заключительные данные в вычислении переплексии представлены в таблице 5.

Таблица 5 – Значения перплексии для моделей LDA, PLSA, ARTM

Модель	Значение переплексии
LDA	7.536
PLSA	8.214
ARTM	236.78

Образец LDA обладает самым низким значением перплексии, однако перечень главных текстов в ней менее понятен. Модели PLSA и ARTM приблизительно схожи по понятности, но всё же ARTM наилучшая верность перплексии. Это обозначает, что хотя темы в PLSA меньше понятны, нежели в LDA, но ARTM добивается наилучшей верности в целом. ARTM располагает несколькими преимуществами:

– модель ARTM учится скорее, да и не просит трудоемких опций и вычислений;

– мы воспользуемся различными способами регуляризации, для настройки модели под наши нужды;
– возможно разрабатывать модели, которые разрешают различные задачи одновременно;
– сокет ARTM элементарен в применении и не требует особых познаний в математике или статистике.

Выводы по главе 3:

В тематическом моделировании имеется множество методом и алгоритмов. Каждый из них подходит для определенных ситуаций. Например, один алгоритм может существовать вернее для анализа комментариев в социальных сетях, а другой - для анализа академических заметок либо новостей.

Из итогов изучения алгоритмов тематического моделирования мы выяснили следующее:

– в общем, темы в модели ARTM в таком роде понятны, как и в других методах, однако они более разнообразны;
– LDA наиболее устойчива к шуму и универсальна, нежели PLSA;
– модель LDA представила наилучшее значение перплексии, в сравнении с PLSA и ARTM;
– темы в PLSA менее понятны, нежели в LDA, потому что PLSA не предусматривает связи среди слов;
– для обрабатывания огромных размеров текста ARTM вызывает побольше вычислительных ресурсов;
– ARTM сможет являться полезным, исключительно при исследовании с разнообразными видами информации либо управлении качеством модификации с через регуляризацию;
– ARTM предоставляет ресурс контролировать разные аспекты моделирования, подобным разреженности тем, коллективное моделирование тем и метаданные;

– ARTM связывает достижения LDA и PLSA, начиная с регуляризации для повышения верности модели, что делает его более эластичным и четким методом.

Избрание среди LDA, PLSA и ARTM зависит от ваших определенных необходимостей и данных.

Когда необходима точность и работа с большим размером текста, то LDA будет наилучшим выбором.

Если необходима огромная верность и учет приближения слов, то PLSA возможно будет являться полезным. Если вы ищете универсальность и лучшую точность в обработке данных, так ARTM сможет быть преимущественным видом.

Также в конечном разделе ВКР было создано и протестировано программное обеспечение. Применили библиотеку Beautiful Soup, предоставляющий инструментарий Python.

Заключение

Бакалаврская работа посвящена исследованию и реализации алгоритмов анализа новостей в информационном пространстве.

Цель работы – предоставить информацию о разработке исследовательских и внедренческих методологий для алгоритмов анализа новостей с целью улучшения понимания и обработки новостных данных.

Это демонстрируется на примере разработки и внедрения конкретных алгоритмов, направленных на повышение точности и эффективности анализа новостей.

В ходе выполнения ВКР были поставлены задачи на исследования.

Получено задание исследовать, как найти информацию в интернете, и проанализировать различные методы сбора и анализа новостей. Мы также провели сравнительный анализ различных подходов, чтобы выявить их сильные и слабые стороны.

Дальше мы описали, как мы анализируем новостную информацию и отображаем наиболее актуальные новости.

Был произведен обзор алгоритмов тематического моделирования для анализа новостей в информационном пространстве, описали и реализовали алгоритмы анализа новостей.

Мы также придумали, как должна работать программа, для возможности разработки программного обеспечения для неё.

В заключительной главе мы разработали и испытали программное обеспечение. Для этого мы использовали библиотеку Beautiful Soup, которая представляется частью Python. Затем, после написания программы мы провели её тестирование.

Выполнена оценка эффективности разработанного программного обеспечения.

Были выполнены все задачи, необходимые для достижения целей работы.

Список используемой литературы и используемых источников

1. Ачкасова В.А. Связи с общественностью как социальная инженерия / В.А. Ачкасова, Л.В. Володина. – СПб.: Речь,2005.–336с.
2. Барткевич Е.А. Продвижение в интернет-сообществе / Е.А. Барткевич. – М.:Erstmedia,2015.–125с.
3. Бочаров М.П. Связи с общественностью. Теория и практика / М.П. Бочаров, А.Н. Чумиков. – М.:Дело,2015.–552с.
4. Брассингтон Ф., Петтитт С. Основы маркетинга / Ф. Брассингтон, С. Петтитт.– М.: Баланс Бизнес Букс, 2014.–536с.
5. Гавра Д.П. Основы теории коммуникации: Учебное пособие. –СПб.: Питер,2014. –288с.
6. Горбачев М.Н. Дистрибуция и продвижение продукта на рынке. Практическое руководство. / М.Горбачев, Я. Газин. – Ростов-на-Дону: Феникс,2014. –159с.
7. Интегрированные коммуникации. Основы рекламы и связей с общественностью: учебное пособие /В.А. Барежев, И.А. Быков ,М.В. Гончаренко и др.; подред. А.Д. Кривоносова.– СПб.: Изд-во СПбГЭУ,2014. – 170с.
8. Калитка О.В. Как сделать корпоративный блог интересным? Стратегия развития и наполнение контентом / О.В. Калитка // Маркетинговые коммуникации.– М.: дом Гребенникова,2014. –№3.– С.140-147
9. Керпен Д., Маркетинг эпохи Like. Как найти и удержать клиентов, создать узнаваемый бренд и нравиться в фейсбуке и других социальных сетях: Пер.Е. Фотьянова.–М.: ШКИМБ,2015.–240с.
10. Кривоносов А.Д. Основы теории связей с общественностью. / А.Д. Кривоносов, О.Г. Филатова, М.А. Шишкина.–Спб.: Питер,2014. –384 с.
11. Кузнецов П.А. Связи с общественностью для бизнеса: практические приемы и технологии /П.А. Кузнецов.– М.:ДашковиКо,2015. –296с
12. Мазилкина Е.А. Семь шагов к успеху или как продвинуть свой товар

на рынок. – Саратов: Ай Пи Эр Медиа, 2012.– 160 с. 21. Мазилкина Е.А. Условия успешного продвижения товара: Практическое пособие – 2-е изд. – М.: Дашкови Ко,2015.– 172с.

13. Маркетинг: Большой толковый словарь./А.П. Панкрухин; Подобщ. Ред. А.П. Панкрухина. Издано при поддержке Гильдии маркетологов. – 3-еиздание – М.:Омега-Л,2015.–264 с.

14. Марков А.А. Теория и практика связей с общественностью: учебное пособие / А.А.Марков.– СПб.:СПбГИЭУ,2015. –163 с.

15. Менцев М. i-SMM Эффективный маркетинг в Instagram [Электронный ресурс] // Slide Share. – URL: http://slideshare.net/art23/instagram-43285231.– (дата обращения: 02.02.2023).

16. Никифорова Л.Х. Подходы к оценке эффективности систем мотивации персонала [Электронный ресурс] // Экономика и менеджмент инновационных технологий. – 2016. – № 6. – URL: http://ekonomika.snauka.ru/2016/06/11632.– (дата обращения: 02.02.2023).

17. Основы маркетинга, 5-е европейское изд.: Пер. с англ. / Ф. Котлер Ф. Г. Армстронг, В. Вонг, Д. Сондерс. – М.: ООО «И.Д. Вильямс», 2016. – 752с.

18. Продвижение в социальных сетях: задачи SMM для государства [Электронный ресурс] // SMM для государственных структур. URL:http://domdruzei.ru/index.php?showtopic=14816. – (дата обращения: 02.02.2023).

19. Рудская Е. Н. Шоу-румы как инструмент интеграции онлайн и офлайн торговли: универсализация каналов продвижения и продаж / Е.Н. Рудская, Е.М. Лобзенко // Молодой ученый. –2014. –№20. –С.396-402.

20. Сосновский С. Эффективное продвижение в социальных сетях [Электронный ресурс] // Сосновский.ру.–URL:http://sosnovskij.ru.– (дата обращения: 02.02.2023).

21. Успешный контент-маркетинг [Электронный ресурс] //Rusability. – URL: https://rusability.ru/content-marketing/8-primerov-uspeshnogo-kontentmarketinga.– (дата обращения: 02.02.2023).

22. Федеральный закон «О защите конкуренции» от 26.07.2006 N 135-ФЗ (последняя редакция)

23. Федеральный закон «О рекламе»от13.03.2006N38-ФЗ (последняя редакция)

24. Халилов, Д. Маркетинг в социальных сетях. – 2-е изд., перераб. И доп..– М.: Манн,ИвановиФербер,2014.–240с.

25. Яцюк Н. Анатомия сарафанного маркетинга.–М.:Манн, Иванов и Фербер,2015. –416с.

26. Instagram: эффективное продвижение от А до Я.–М.:Ingate,2014 – 35с.

27. Key Social Media Metrics [Электронный ресурс]// blog.–URL: https://blog.bufferapp.com/social-media-metrics.–(дата обращения:02.02.2023)

28. Charleis H. Granger. The Hierachy of Objectives. Harvard Busines Review, May-June 2017,56– 70.

29. Komornik, Vilmos. Fourier series in control theory / Vilmos Komornik, Paola Loreti.-[NewYorketc.]:Springer,cop.2015.-226c.

30. Sheth, Jadish. Internet marketing / JagdishN. Sheth, Abdolreza Eshghi, Balaji C.Krishnan.-Fort Worth:Harcourt college,2016.– 419.

31. Wind, Jerry; Mahajan, Vilay. Digital marketing: Global strategies from the world are leading experts, cop.2017.