Санкт–Петербургский государственный университет

Тарасов Никита Андреевич

Выпускная квалификационная работа

Суммаризация мультиязычных дискуссий в социальных сетях

Уровень образования: магистратура
Направление 02.04.02 «Фундаментальная информатика и информационные технологии»
Основная образовательная программа «Технологии баз данных»
№19/5503/1

Научный руководитель:
Кандидат технических наук, Доцент,
Заведующий кафедрой технологии программирования
Блеканов Иван Станиславович

Рецензент:
Начальник отдела стратегических инициатив
Центр разработки и монетизации данных, Газпром Нефть
Кононов Ярослав Сергеевич

Санкт-Петербург
2021 г.

Saint–Petersburg State University

Tarasov Nikita Andreevich

Graduation project

Multingual discussion summary in social networks

Level of education: Master level
Main field of study 02.04.02
«Fundamental Informatics and Information Technology»
Academic programme title: «Database Technologies» №19/5503/1

Research supervisor:
Ph.D., Associate Professor,
Head of department of programming technologies
I.S.Blekanov

Reviewer:
Head of Strategic Initiatives Department
Center of Development and Monetization, Gazprom Neft
Yaroslav Kononov

Saint–Petersburg
2021 г.

Оглавление

Введение 4
1. Актуальность 4
2. Цель работы 5
3. Задачи 6
4. Значимость 7

Глава 1. Обзор методов и инструментов анализа 8
1.1. Методы кодирования данных 8
1.1.1. Bag of Words 9
1.1.2. TF-IDF 9
1.1.3. Word2vec 10
1.1.4. FastText 11
1.1.5. ELMo 12
1.1.6. Transformer 13
1.1.7. Universal Sentence Encoder 16
1.1.8. BERT 17
1.1.9. T5 20
1.1.10. LongFormer 23
1.2. Тематическое моделирование 26
1.2.1. Latent Dirichlet Allocation 26
1.2.2. Word Network Topic Model и Biterm Topic Model . . 27
1.2.3. Bilingual Latent Dirichlet Allocation 28
1.2.4. Нейросетевые подходы 29
1.3. Методы суммаризации 32
1.3.1. Экстрактивная суммаризация 32
1.3.2. Абстрактная суммаризация 33
1.3.3. Применение языковых моделей 33
1.4. Методы анализа тональности 37
1.4.1. Традиционные подходы 37
1.4.2. Подходы на основе нейросетей 38
1.5. Анализ коммерческих инструментов 39
1.5.1. IBM Watson 39

1.5.2. MeaningCloud 40

Глава 2. Система анализа пользовательских дискуссий 42
2.1. Архитектура комплекса 42
2.2. Данные - сбор и анализ 44
2.3. Кластерный анализ 48
2.4. Тематическое моделирование 50
2.5. Суммаризация дискуссий внутри отдельных постов 52
2.6. Суммаризация и тональность временных рядов 54

Заключение 56
Результаты работы 56
Перспективы развития 58

Список литературы 59

Введение

1. Актуальность

На сегодняшний день интернет играет важную роль в социально и политической жизни общества. Объем данных в интернете продолжает возрастать экспоненциально. Немалая часть общего числа данных приходится на текстовые данные. К ним относятся как авторский контент: записи в блогах, новостные и научные статьи, так и обсуждения различных тем: комментарии к новостям, видео, товарам. Так, например, все большее число людей отдает предпочтение интернет СМИ и социальным медиа. Соцсети при этом занимают важную роль, так как имеют большой охват пользователей, часто оказываются первоисточником информации о крупных социальных, экономических и политических событиях, а также являются удобными платформами для дискуссий. Так в исследованиях statista [1] 2020 года приводится информация о высокой скорости роста числа пользователей соцсетей - 3.6 млрд. пользователей на 2020 год, что соответствует 34 процентному приросту по сравнению с 2017 годом. Они нередко становятся основной площадкой обсуждения этих событий, аккумулируя мнения сотен различных людей. В связи с невозможностью ручной обработки больших объемов данных, все более актуальной становится задача анализа текстовых данных, полученных из различных источников. Подобный анализ позволяет не только получить наиболее полную и качественную информацию в максимально короткий срок, но также представить ее в виде, максимально доступном и удобном для дальнейших исследований. В данной теме можно выделить большое количество различных подзадач: суммаризации, анализ тональности, ответы на вопросы, машинный перевод и др. Существует также и немалое число способов для их решения - от простейших подходов, основанных на правилах и статистики встречаемости слов, до современных подходов, основанных на языковых моделях, способных решать одновременно множество задач, требуя при этом значительно меньшего участия специалистов по разметке данных.

2. Цель работы

В данной работе автором рассматриваются различные подходы к суммаризации текстовых данных, полученных из социальных сетей. В качестве базы производится анализ методов тематического моделирования, применение которых позволяет как уменьшить размерность данных, так и выделить основные идеи, аргументы и мнения с учетом специфики начальных данных. К особенностям данных, полученных из соцсетей можно отнести низкую среднюю длину слов так как зачастую мнения выражаются тезисами, а не длинными предложениями. Также важную роль играет мультиязычность, так как крупные общественные события привлекают внимание не только носителей местного языка, но и иностранцев.

Основной целью работы является рассмотрение современных подходов, основанных на языковых моделях, использование которых позволяет решить проблемы обозначенные выше, не прибегая к специфическим модификациям традиционных моделей. Сравнение данных подходов с классическими позволит выделить функции и области, с которыми подходы моделирования языка справляются лучше, выявить их ограничения.

Дополнительно для языковых моделей рассматривается возможность использования результатов методов суммаризации текстовых данных совместно с методами анализа определения тональности. Данный раздел анализа естественного языка позволяет получать информацию об эмоциональном окрасе высказываний, что актуально как в задаче исследования социальных дискуссий, так и в задаче анализа отзывов на определенные товары.

Использование алгоритмов выявления ключевых фрагментов и методов анализа тональности совместно с методами статистического анализа данных, позволят не только эффективно суммаризировать пользовательские дискуссии, но и получать значимые параметры и выборки данных.

3. Задачи

Обозначенные цели требуют всестороннего анализа литературы, методов и инструментов. Для практической части необходимо разработать архитектуру программного комплекса для анализа данных, полученных из соцсетей и показать примеры обработки корпусов из различных источников. Следовательно, к числу основных задач данной работы можно отнести:

1. Рассмотрение основных методов обобщения текстов
2. Анализ основных методов тематического моделирования
3. Описание алгоритмов работы и основных преимуществ нейросетевых методов
4. Разработка архитектуры программного комплекса
5. Сбор данных и их предварительная обработка
6. Реализация и тестирование методов
7. Проведение сравнительного анализа методов
8. Визуализация полученных результатов

4. Значимость

Решение обозначенных задач актуально для различных сфер и имеет целый ряд применений как коммерческих, так и социальных. К числу ключевых направлений можно отнести анализ мнений потребителей о товарах для выявления ключевых достоинств и проблем, выявление общественного мнения по поляризованным темам. В случае анализа товаров и услуг, всесторонний анализ пользовательских мнений позволяет маркетологам быстрее и качественнее оценивать рынок, конкурентов, стратегии продвижения. В анализе мнений общественности на острые социальные и политические темы, выявление общих тем и тональности дискуссий может значительно упрощать работу экспертов в области социологии, политологии и журналистики. Примерами реальных случаев, когда необходим подобный анализ, могут быть:

- Крупные беспорядки - деэскалирование ситуации, определение мотивов и целей различных групп
- Выпуск нового товара - анализ рынка, выявление ключевых достоинств и недостатков выпускаемого продукта
- Природные и техногенные катаклизмы - повышение осведомленности среди населения
- Важные политические события (выборы, законопроекты) - анализ мнений избирателей

В данных примерах работа профессиональных специалистов улучшается как за счет качественной агрегации данных (выборки данных по времени, по авторам, сообществам), так и непосредственно с помощью методов суммаризации и анализа тональностей, позволяющих оценивать крупные выборки через их общие краткие представления в совокупности с понятными представлениями общего отношения пользователей к той или иной теме.

Глава 1

Обзор методов и инструментов анализа

В течении последних десятилетий было предложено множество способов суммаризаций текстовых данных, начиная от использования общих методов кластеризации, заканчивая специфическими нейросетевыми методами. Методы кластеризации позволяют, используя различные методы представления текстовых данных, группировать тексты в блоки, имеющие общее содержание [2]. Особое значение в данном, базовом варианте группировки данных, занимает способ кодирования исходных текстов.

1.1. Методы кодирования данных

Далее приведены способы кодирования текстовых данных, используемых моделями суммаризаций и тематического моделирования. Модели приведены в порядке усложнения архитектуры до модели Transformer. Дальнейшие модели являются различными ее вариациями, предназначенными и оптимизированными для решения различных задач. Использование более сложных современных подходов обусловлено в первую очередь возможностями практического применения методов кластеризации, тематического моделирования и суммаризации. Все эти методы, так или иначе, ставят уменьшение объема текстовых данных без потерь качества, как одну из основных задач. Дополнительно, в случае тематического моделирования, полученные тематические представления документов можно использовать как кодировки для дальнейших задач, таких как анализ тональности, перевод, ответы на вопросы и других задач анализа естественного языка. Однако, как будет показано в дальнейшем, данные задачи решаются языковыми моделями путем получения последнего скрытого состояния модели. Данный подход позволяет получать качественные представления данных, оптимизированных для специфичных задач [3].

1.1.1. Bag of Words

Bag of Words (Bow) является одним из наиболее распространенных способов представления текстовых данных [4]. В данном подходе создается матрица $D \times W$, где D - число предложений в корпусе, а W - размер словаря корпуса. Элемент D_i, W_j данной матрицы таким образом указывает количество вхождений слова j в предложение i. Основные достоинства данного метода:

- Простота реализации – реализован почти во всех фреймворках работы с естественным языком
- Скорость работы – простейшие операции подсчета слов
- Разряженные матрицы позволяют использовать специальные форматы хранения данных

Ключевыми недостатками данного подхода являются:

- Невозможность учитывать порядок слов
- Ненормированность итоговой матрицы - большой вес отдается частым словам (стоп словам и общим терминам)
- Необходимость приведения слов к начальной форме и как следствие - невозможность учитывать словоформы

1.1.2. TF-IDF

Произведение количества вхождений слова в предложение (частота термов, TF) на значение усиливающее вклад редких слов (обратная частота документов, IDF) [5]. Для кодирования отдельных слова t в документе d при этом чаще всего используется следующее выражение:

$$\underbrace{(0.5 + 0.5\frac{f_{t,d}}{\max_t f_{t,d}})}_{TF} * \underbrace{\log \frac{N}{n_t}}_{IDF}$$

где $f_{t,d}$ – базовое число вхождений слова в документ (аналогично BOW), N – число документов в корпусе, n_t – число документов, в которых встречается данное слово.

Данный подход позволяет уравновешивать слова, часто встречающиеся в общем случае, автоматически уменьшая веса стоп-слов, уменьшая необходимость в предобратке.

Основные преимущества данного подхода:

- Простота использования
- Высокая скорость работы
- Нормализован - частые слова имеют штрафной коэффициент

Основные недостатки:

- Невозможность учитывать порядок слов
- Невозможно учитывать словоформы, семантику
- Недостаточно прочная теоретическая основа - высокие значения метрики для отдельных слов в предложениях не всегда коррелируются с их темами

1.1.3. Word2vec

Word2vec [6] - это двухслойная нейронная сеть, обрабатывающая текст путем «векторизации» слов. Его входными данными является текстовый корпус, а его выходными данными - набор векторов: векторов признаков, которые представляют слова в этом корпусе.

Word2vec похож на автоэнкодер, кодирующий каждое слово в векторе, но вместо обучения по входным словам посредством реконструкции, как это делает ограниченная машина Больцмана, word2vec тренирует слова по другим словам, которые соседствуют с ними во входном корпусе.

Это делается одним из двух способов: либо с использованием контекста для предсказания целевого слова (метод, известный как непрерывный мешок слов, или CBOW), либо с использованием слова для предсказания целевого контекста и называется skip-gram.

Основными достоинствами данного метода являются:

- Простая и интуитивно понятная идея
- Возможность кодирования похожих слов близкими представлениями

Недостатки подхода:

- Низкая скорость работы при больших размерах словаря
- Не может работать со словами, ранее не встречающимися в словаре
- Репрезентации слов не зависят от контекста (например слово "замок"всегда кодируется одинаково, несмотря на разные значения в конкретных предложениях)

1.1.4. FastText

FastText [7] позволяет эффективно кодировать текстовую информацию, путем представления слов векторами определенной размерности. Идея и реализация данной модели схожа с классической моделью word2vec, однако имеет перед ним ряд преимуществ. Основным достоинством модели по сравнению с другими подходами моделирования языка, является выражение слов через составляющие их n-граммы. Это позволяет использовать более глубокую информацию символьного уровня, в отличие от классических методов кодирования таких как bag of words, tf-idf, word2vec. Данный подход позволяет модели лучше работать с короткими текстами, позволяет распознавать части слов, а также кодировать новые слова, не встречающиеся в текстовом корпусе. FastText поддерживает обучение методом Continous Bag of Words (CBOW) или модели Skip-gram с использованием негативного сэмплирования, softmax или иерархической softmax функции потерь. Использование негативного сэмплирования и модифицированной функции потерь позволяет получать качество моделей, близкое или превосходящее качество схожих моделей глубокого обучения, используя при этом заметно меньшие вычислительные ресурсы и затрачивая на обучение и кодирование значительно меньшее время. Авторы оригинальной статьи приводят множество различных предобученных моделей для разных языков, обученных

на комбинированных коллекциях текстовых данных из открытых источников.

Основные достоинства:

- Структура схожая с word2vec - доказана высокая эффективность и интуиция метода
- Использование новых идей модификации (например негативного сэмплирования) позволяет существенно ускорить метод
- Работа с n-граммами позволяет кодировать слова не встречающиеся в корпусе, различные формы одного слова

Основные недостатки:

- Невозможность кодировать контекст слов - аналогично модели word2vec

1.1.5. ELMo

ELMo [8] – это модель, создающая глубокие контекстуализированные представления слов. Данный алгоритм моделирует как сложные характеристики использования слов (например, синтаксис и семантику), так и то, как эти качества различаются в зависимости от языкового контекста (например, при моделировании многозначности слов). В отличие от других широко используемых текстовых эмбеддингов, представления слов в ELMo являются функциями всего входного предложения, а не отдельных слов.

Состоящие из одной прямой и одной обратной языковых моделей, скрытые состояния ELMo имеют доступ как к предыдущему, так и к следующему в сообщении словам. Каждый скрытый слой является двунаправленным LSTM, поэтому данная языковая модель способна различать скрытые состояния с любого направления. После обучения прямой и обратной языковых моделей ELMo объединяет веса скрытых слоев в единое представление.

Итоговую модель обучения biLM (двунаправленных языковых моделей) ELMo на большом текстовом корпусе, можно использовать аналогично с результатами модели FastText, с тем отличием, что результирующая кодировка не будет уникальна для каждого слова, а будет зависеть от контекста, т.е. окружающих его слов.

Основные достоинства:

- Контекстная модель
- Использование LSTM позволяет эффективно захватывать информацию о прошлых словах

Ключевые недостатки:

- Использование LSTM ограничивает параллелизацию
- Модели, работающие на уровне символов, показывают худшие результаты по сравнению с моделями уровнями слов и n-грамм [9]
- Захват контекста только в одном направлении

1.1.6. Transformer

Предобученные языковые модели стали ключевым элементом в достижении state of the art результатов во многих задачах обработки текстовых данных [10]. Данные модели расширяют идеи кодирования слов рассмотренные ранее, главным образом за счет изучения и получения контекстуальных репрезентаций из больших наборов данных, используя некоторую целевую задачу моделирования языка. Рассматриваемые ранее модели NVDM и ELMo являются примерами первых универсальных языковых моделей, применение которых позволило получить лучшие на свое время (2019 год) результаты по целому ряду задач анализа текстовых данных. Однако на данный момент их применение оправдано лишь в редких случаях (например невозможности распараллеливания анализа данных по техническим причинам), ввиду большого числа преимуществ, предоставляемых более современными подходами.

Transformer был разработан как архитектура преобразования последовательностей. Примерами задач, где входная последовательность преобразуется в выходную могут служить машинный перевод, распознавание речи, суммаризации и многие другие области. Традиционно данные задачи решались рекуррентными нейросетями [11] и их модифицированными версиями на основе долгой краткосрочной памяти (Long-Short Term Memory) [12].

В базовой версии рекуррентная сеть состоит из двух компонентов: кодировщика и декодировщика. Обе этих части работают как стандартные рекуррентные сети, то есть используют на входе не только информацию из ввода, но и информацию из предыдущих выходов модели. При этом в отличии от простейшего случая применения рекуррентной сети, использование двух сетей позволяет обойти ограничение на равенство длины входа и выхода модели и порядка слов, то есть нет необходимости в прямом соответствии входных данных и целевых значений. Кодировщик в данном случае получает вектор и скрытое состояние, которое затем используется совместно с вектором кодировки следующего слова. Декодировщик, в свою очередь, получает на вход кодировки и выводит последовательность слов. Основным недостатком данного подхода является невозможность параллелизации вычислений, так как для каждого следующего слова требуется скрытое состояние кодировщика предыдущего слова. Также большой проблемой является невозможность учитывать связи слов в больших предложений (алгоритм отдает предпочтение ближайшим предыдущим словам). LSTM модифицирует данную структуру с помощью ячеек состояния. Данные ячейки позволяют добавлять или удалять информацию с помощью, так называемых, гейтов. Гейты позволяют частично пропускать информацию и состоят из сигмоидного слоя и операции поэлементного умножения. Подобный механизм выборочной памяти позволяет лучше улавливать долгосрочные связи в предложениях, однако имеет все те же проблемы с параллелизацией.

Transformer модель решает проблемы стандартных подходов, используя архитектуру, полностью построенную на механизме внутреннего внимания (self-attention) [13]. На Рис 1.1. представлена архитектура данной модели. Слева на данной схеме изображен кодировщик, а справа декодировщик, при этом данные блоки могут дублироваться k раз (в оригинальной статье используется 6 последовательных блоков). Основными компонентами кодировщика и декодировщика являются модули внутреннего внимания и слои прямого распространения (Feed Forward Layers). Входы и выходы (целевые предложения) сначала встраиваются в n-мерное пространство. При этом важная часть модели - позиционное кодирование разных слов. Поскольку данная модель не использует рекуррентные сети, которые могут

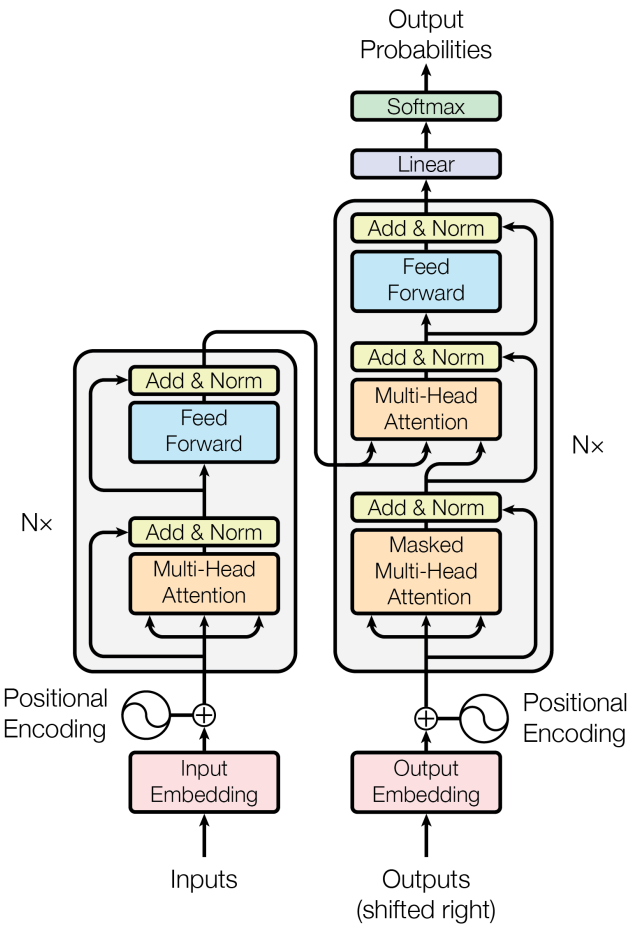

Рис. 1.1. Архитектура модели Transformer

запомнить, как последовательности слов используются в модели, возникает необходимость присвоить каждому слову относительное положение, поскольку последовательность зависит от порядка ее элементов. Эти позиции добавляются к встроенному представлению (n-мерному вектору) каждого слова. Функцию внутреннего внимания при этом можно описать как отображение входа и набора пар ключ-значение на выход, где запрос, ключи, значения и выходные данные являются векторами. Выходные данные вычисляются как взвешенная сумма значений, где вес, присвоенный каждому значению, вычисляется функцией совместимости запроса с соответствующим ключом.

Основные преимущества модели:

- Улучшенное моделирование дальних зависимостей слов
- В отличии от прошлых подходов на основе RNN, возможна параллелизация методов

Основные недостатки:

- Сложность предобучения seq2seq моделей, т.к. архитектура состоит из двух частей
- Механизм внимания работает с строками фиксированной длины, как следствие требует разбиений, что в некоторых случаях ведет к потере контекстной информации

1.1.7. Universal Sentence Encoder

Данная модель кодирует текст в многомерные векторы, которые можно использовать для классификации текста, определения семантического сходства, кластеризации и других задач обработки естественного языка [14]. Ее отличительными особенностями являются высокое качество кодировки целых предложений и возможность использования одного из двух типов архитектуры. Один, основанный на ранее описанной архитектуре transformer, нацелен на высокую точность за счет большей сложности модели и увеличения потребления ресурсов. Другой подход нацелен на эффективный вывод с немного сниженной точностью.

Модель кодирования предложений на основе transformer сети конструирует кодировки предложений с использованием подграфа кодирования архитектуры преобразователя. Этот подграф использует внимание для вычисления контекстно-зависимых представлений слов в предложении, которые учитывают как порядок, так и содержание всех других слов. Представления слов с учетом контекста преобразуются в вектор кодирования предложений фиксированной длины путем вычисления поэлементной суммы представлений в каждой позиции слова. Кодировщик принимает в качестве входных данных токенизированную строку PTB в нижнем регистре и выводит 512-мерный вектор в качестве эмбединга предложения. Модель кодирования разработана так, чтобы быть максимально универсальной. Это достигается с помощью многозадачного обучения, при котором одна модель кодирования используется для выполнения нескольких последующих задач. Авторы показывают что данный вариант архитектуры показывает наилучшее качество, однако требователен как по памяти, так и по времени, особенно в случае кодирования длинных предложений.

Вторая модель кодирования использует Deep Averaging Network (DAN), посредством которой входные эмбединги для слов и биграмм сначала усредняются вместе, а затем проходят через глубокую нейронную сеть прямого распространения для создания кодировок для предложений [15]. Подобно кодировщику Transformer, кодировщик DAN принимает в качестве входных данных токенизированную строку PTB в нижнем регистре и выводит 512-мерные эмбединги предложений. Кодировщик DAN обучается аналогично кодировщику на базе transformer. Основное преимущество кодировщика DAN заключается в том, что время кодирования линейно зависит от длины входной последовательности и как следствие имеет намного более высокую скорость работы и меньшие затраты по памяти, незначительно теряя при этом в качестве получаемых кодировок предложений.

Основные достоинства модели:

- Удобные инструменты работы с моделью
- Возможность выбора архитекутры при обучении
- Большое число предобученных мультиязычных моделей

Ключевые недостатки:

- Односторонний контекст в двух вариантах архитектуры
- Необходимость выбора между качеством и скоростью (Transformer и DAN соответственно)

1.1.8. BERT

Bidirectional Encoder Representations from Transformers продолжил идеи архитектуры трансформер модели, модифицируя часть кодировщика. BERT предназначен для предварительного обучения глубоких двунаправленных представлений из немаркированного текста путем совместной обработки левого и правого контекста на всех уровнях [16]. В результате предварительно обученная модель BERT может быть настроена всего с одним дополнительным выходным слоем для создания качественных моделей для широкого круга задач, без существенных модификаций архитектуры

для конкретных задач - идея, получившая название трансферное обучение и описанная в разделе 1.1.7 для модели Universal Sentence Encoder.

На рис. 1.2. показана архитектура кодировщика модели BERT. Входные данные - это последовательность токенов, которые сначала встраиваются в векторы, а затем обрабатываются стандартной трансформер-сетью. Выходные данные представляют собой последовательность векторов размера H, в которой каждый вектор соответствует входному токену с тем же индексом.

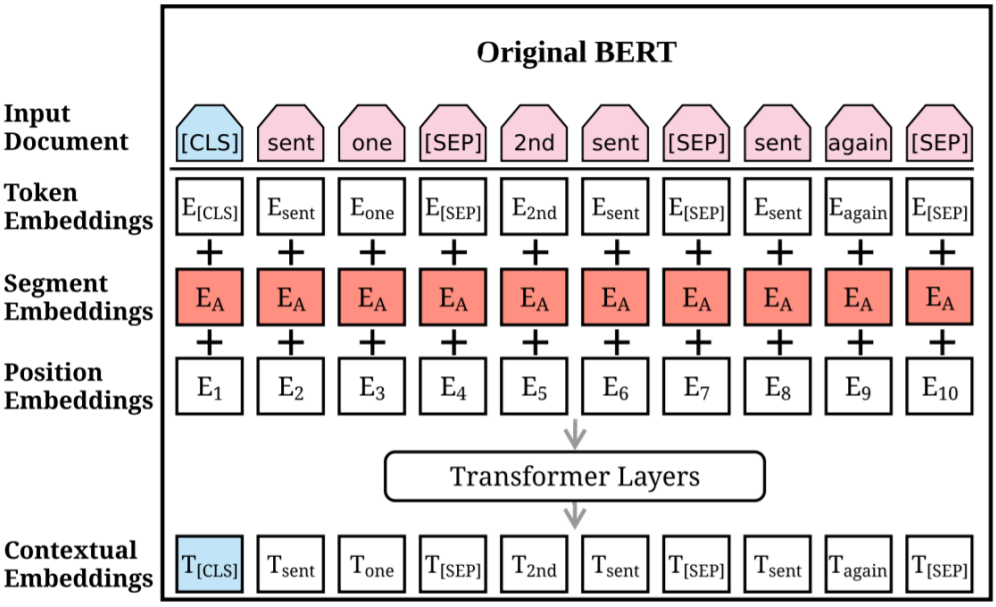

Рис. 1.2. Архитектура модели BERT

При обучении языковых моделей возникает проблема определения цели прогнозирования. Многие модели предсказывают следующее слово в последовательности - это направленный подход, который по своей сути ограничивает контекстное обучение. Чтобы преодолеть эту проблему, BERT использует две различные стратегии обучения.

Masked LM это первая стратегия обучения предобучения модели. Интуитивно разумно полагать, что глубокая двунаправленная модель будет мощнее, чем модель, воспринимающая контекст только слева направо или неглубокая конкатенация моделей, работающих слева направо и справа налево. К сожалению, стандартные модели можно обучать только слева направо или справа налево, поскольку двунаправленное обуславливаение позволит каждому слову косвенно видеть себя, и как следствие, модель может тривиально предсказать целевое слово в многоуровневом контексте.

Чтобы обучить глубокое двунаправленное представление, модель случайным образом маскирует некоторый процент входных токенов, а затем прогнозирует эти замаскированные токены. Хотя такой подход и позволяет получить двунаправленную предварительно обученную модель, недостатком подхода является несоответствие между предварительным обучением и тонкой настройкой, поскольку токен $[MASK]$ не появляется во время точной настройки. Чтобы смягчить это, замаскированные слова не всегда заменяются фактическим токеном [MASK]. Вместо этого генератор обучающих данных случайным образом выбирает 15% позиций токенов для прогнозирования. Если выбран i-й токен, он заменяется токеном $[MASK]$ в 80% случаев, случайным токеном в 10% случаев и остается неизмененным в 10% случаев. Итоговый T_i используется для прогнозирования исходного токена.

Next Sentence Prediction является вторым подходом к предобучению финальной модели. Многие важные последующие задачи, такие как ответы на вопросы (QA) и логический вывод на естественном языке (NLI), основаны на понимании взаимосвязи между двумя предложениями, которые напрямую не фиксируются языковыми моделями. Чтобы обучить модель, которая понимает отношения предложений, модель предварительно обучается бинарной задаче предсказания следующего предложения, которая может быть тривиально сгенерирована из любого корпуса. В частности, при выборе предложений A и B для каждого примера предварительного обучения, в 50% случаев B является фактическим следующим предложением, следующим за A, и в 50% случаев это случайное предложение из корпуса.

При обучении модели BERT, Masked LM и Next Sentence Prediction обучаются вместе с целью минимизировать комбинированную функцию потерь этих двух стратегий. Основные достоинства модели:

- Высочайшие показатели качества модели для большинства задач анализа языка
- Двунаправленная модель, захватывающая контекст слов с двух сторон
- Обучения на двух типах задач позволяет эффективно решать разно-

образный спектр задач и предобучать модели, подходящие для разнообразных данных

Ключевые недостатки:

- Высокие требования к оборудованию (GPU с высокими объемами памяти)

1.1.9. T5

Основная идея, лежащая в основе работы «Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer» [17], состоит в том, чтобы рассматривать каждую проблему обработки текста как проблему «преобразования текста в текст», т.е. принимать текст в качестве входных данных и создавать новый текст в результате работы модели. Подобный подход встречался ранее в других моделях с унифицированными структурами для различных задач НЛП. Примерами могут служить: фреймворк, преобразующий все задачи обработки текста в задачу ответов на вопросы [18] и языковое моделирование [19]. Важно отметить, что структура преобразования текста в текст позволяет напрямую применять одну и ту же модель, цель, процедуру обучения и процесс декодирования к каждой, рассматриваемой задаче, что особенно актуально для рассматриваемой в данной работе цели - всестороннего анализа корпусов пользовательских сообщений. Использование подобной гибкой модели, позволяет получить результат и оценить эффективность для широкого круга задач НЛП, включая обобщение документов (abstarctive summarization) и классификацию настроений (sentiment analysis). На рис. 1.3. показан схематичный пример работы такой модели, получившей название Text-to-Text Transfer Transformer.

Из данного примера можно увидеть как использование такой модели не только упрощает лежащие в основе разных задач идеи, но и позволяет сравнивать эффективность различных целевых функций трансферного обучения, получать качественные результаты на данных, не имеющих предварительной разметки.

В ключевых моментах, данная реализация Трансформер-модели кодировщика-декодера полностью соответствует его первоначально предло-

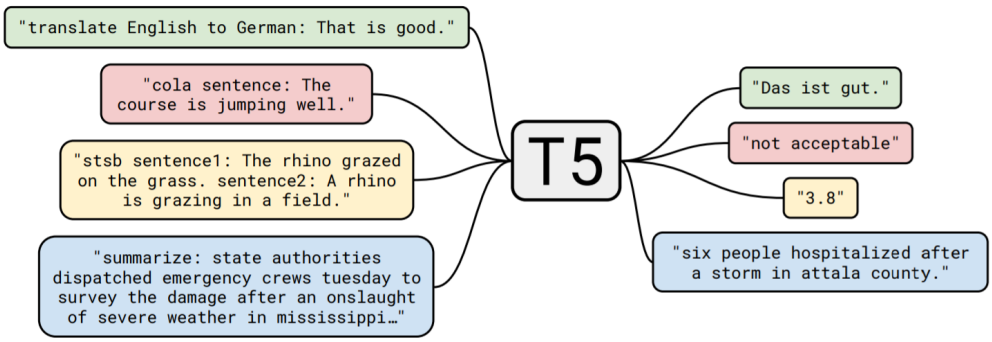

Рис. 1.3. Пример работы T5 модели

женной форме, описанного ранее в данном разделе. Сначала входная последовательность токенов кодируется набором емебедингов, которые затем передаются в кодировщик. Кодировщик состоит из стека «блоков», каждый из которых состоит из двух компонентов: уровня self-attention, за которым следует небольшая нейросеть прямого распространения. Нормализация слоя [20] применяется ко входу каждого компонента. Авторы используют упрощенную версию нормализации слоев, в которой активации лишь масштабируются, без применения аддитивного смещения. После слоя нормализации Residual skip connection [21] добавляет вход каждого подкомпонента к его выходным данным.

Dropout [22] применяется в сети с прямой связью, для пропуска части соединений, в весах слоя внимания, а также на входе и выходе всей структуры. Декодер похож по структуре на кодировщик, за исключением того, что он включает стандартный механизм внимания после каждого слоя самовнимания, который обслуживает выходные данные кодера. Механизм самовнимания в декодере также использует форму авторегрессии или причинного самовнимания, которое позволяет модели обращать внимание только на прошлые результаты. Выходные данные последнего блока декодера передаются в плотный слой с выходом softmax, веса которого разделяются с входным набором эмбедингов. Все механизмы внимания в Transformer разделены на независимые части, выходы которых объединяются перед дальнейшей обработкой.

Одним из ключевых достижений авторов оригинальной статьи стал сбор и подготовка данных для обучения модели. Большая часть предыдущей работы по трансферному обучению для НЛП использовала большие неразмеченные наборы данных для обучения без учителя. Чтобы создать

наборы данных, авторы использовали Common Crawl в качестве источника текста, извлеченного из Интернета. Common Crawl ранее использовался в качестве источника текстовых данных в различных задачах NLP, например, для обучения языковой модели n-грамм [23], в качестве обучающих данных в задаче "анализа здравого смысла-[24], для интеллектуального анализа данных, сбора параллельных текстов в задаче машинного перевода [25], как набор данных для предварительного обучения, и просто как большой текстовый корпус для тестирования оптимизаторов [26]. Common Crawl - это общедоступный веб-архив, который предоставляет извлеченный из Интернета текст с удалением мета-информации и другого нетекстового содержимого из очищенных файлов HTML. Этот процесс производит около 20 ТБ очищенных текстовых данных каждый месяц. К сожалению, большая часть результирующего текста не является естественным языком. Вместо этого он в основном состоит из шаблонных текстов, таких как меню, сообщения об ошибках или повторяющиеся тексты. В связи с этим авторы оригинальной статьи проводят тщательный процесс предварительной обработки, включающий в себя:

- Удаление дубликатов
- Удаление нецензурных слов
- Удаление текстов, относящихся к программному коду
- Отбор текстов на английском языке (с помощью библиотеки langdetect)
- Удаление текстов с сообщениями об ошибках Javascript и иными стандартными кодами ошибок
- Отбор предложений длиной более 3х слов и текстов содержащих более 5 предложений

Чтобы собрать базовый набор данных, был загружен Common Crawl текст за апрель 2019 года с последующим применением описанной выше процедуры фильтрации. В результате получается набор текста, который не только на несколько порядков больше, чем большинство наборов данных, используемых для предварительного обучения (около 750 ГБ), но также содержит достаточно чистый и естественный текст.

Используя данную систему сбора данных в совокупности с одной из лучших архитектур обучения seq2seq моделей, позволило авторам добиться высочайших результатов на нескольких бенчмарках, в частности получить результат максимально приближенный к человеческому в системе оценки качества моделей понимания языка SuperGLUE [27].

Основные достоинства модели:

- Унифицированная структура для всех seq2seq моделей
- Большой и качественный корпус, используемый для предварительного обучения
- Удобная структура тюнинга и получения результатов модели

Ключевые недостатки:

- Модель не получает лучшие результаты в задаче перевода, т.е. качество модели в некоторой степени зависит от языка текста
- В случае задачи классификации, например анализа тональности, от модели ожидается вывод одного из вариантов (например "позитивный"или "негативный"), однако отмечаются случаи, когда модель выводит вывод не из числа вариантов обучающей выборки (такие случаи будут ошибкой модели ввиду ее универсальной seq2sqe структуры)

1.1.10. LongFormer

Модели на основе трансформеров не могут обрабатывать длинные последовательности из-за их работы с самовниманием, алгоритм которого масштабируется квадратично с длиной входной последовательности. Longformer позволяет обходить эти ограничения с помощью механизма внимания, который линейно масштабируется с длиной последовательности, что упрощает обработку документов, состоящих из тысяч токенов и более [28]. Механизм внимания Longformer представляет собой прямую замену стандартного самовнимания и сочетает в себе локальное оконное внимание с глобальным вниманием, мотивированным отдельными задачами. Учитывая важность локального контекста [29], паттерн внимания данной модели использует

окно внимания фиксированного размера, окружающее каждый токен. Использование нескольких составных слоев такого оконного внимания приводит к большому восприимчивому полю, где верхние слои имеют доступ ко всем входным местоположениям и имеют возможность создавать представления, которые включают информацию по всему входу, аналогично CNN [30]. При фиксированном размере окна w каждый токен обслуживает $\frac{1}{2}w$ токенов с каждой стороны. Вычислительная сложность этого алгоритма составляет $O(n \times w)$, то есть масштабируется линейно с размером входной последовательности n.

Для дальнейшего увеличения воспринимающего поля без увеличения вычислений скользящее окно может быть расширено. Аналогичная операция выполняется расширенным CNN [31], где в окне есть промежутки с расширением размера d. Предполагая фиксированные d и w для всех слоев, принимающее поле равно $l \times d \times w$, которое может достигать десятков тысяч токенов даже при малых значениях d. В современных моделях, таких как ранее описанный BERT, для задач естественного языка оптимальное представление ввода зависит от задачи. В данном случае ограниченное и расширенное внимание не является достаточно гибким, чтобы изучать представления для конкретных задач. Соответственно, авторами добавляется механизм общего внимания к нескольким заранее выбранным местам ввода. Важно отметить, что эта операция внимания создается в симметричной форме: то есть токен с глобальным вниманием обслуживает все токены в последовательности, а все токены в последовательности - его.

Результирующая модификация архитектуры Transformer позволяет эффективно захватывать порядок и смысловое содержание слов на протяжении больших документов. На рис 1.4. приводятся следующие показатели эффективности для различных модификаций полученной модели. На данных графиках представлена скорость работы и эффективность по памяти для случая полного слоя внимания и различных имплементаций слоя внимания модели Longformer. По ним видно что в случае стандартного слоя внимания и память и время растут квадратично с ростом длины входной последовательности, тогда как для данной модели во всех модификациях рост линеен.

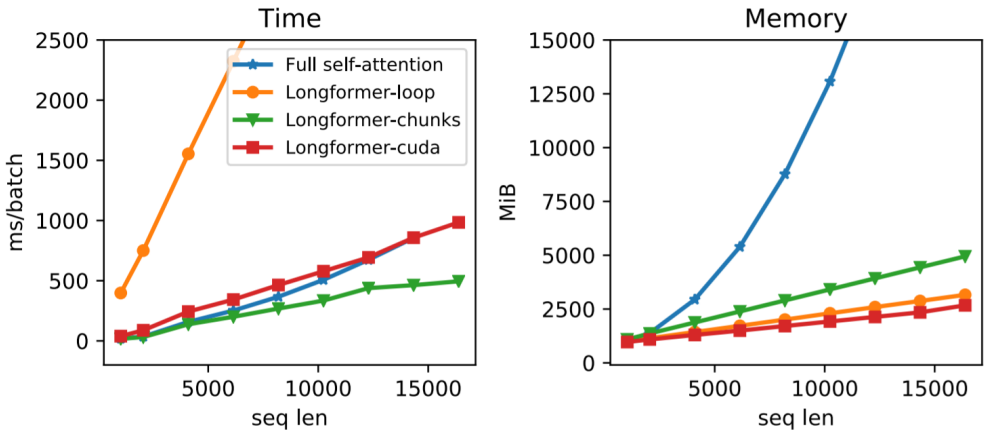

Рис. 1.4. Сравнение скорости работы и потребления памяти для longformer

Основные достоинства модели:

- Решение проблемы больших данных модели Transformer
- Уменьшение требований к объему видеопамяти
- Возможность модификации других методов на основе Transformer

Ключевые недостатки:

- Неполносвязная система внимания ведет к потерям качества, хоть и незначительным
- Специфичная структура оправдывает применения лишь в редких случаях: большие тексты и агрегированные данные

1.2. Тематическое моделирование

Следующим этапом развития моделей суммаризации и описания текстовых данных можно считать тематическое моделирование. Для данной задачи характерна общая постановка и основные определения. Основной задачей является получение репрезентаций текстов из корпуса и тем, образующих эти документы. Таким образом, документы представляются распределение тем, тема - распределение слов. В задаче анализа текстов, полученных из социальных сетей, особый интерес представляют модели, созданные для анализа коротких текстов. Среди подобных моделей анализа коротких и несбалансированных текстов, в работах прошлых лет (ссылки на меня) были выделены следующие основные подходы.

1.2.1. Latent Dirichlet Allocation

В LDA [32] полагается наличие k латентных тем, согласно которым генерируются документы, а каждая тема представляется как мультиномиальное распределение по V словам в словаре. Документ генерируется сэмплированием по этому набору тем, а затем сэмплированием по словам из этого набора. Данный процесс и является постановкой задачи, описанной выше. Более ранние модели работали лишь с одним основным распределением - слов по темам, каждый документ представлялся при этом одной темой. Прочная теоретическая основа, высокие показатели качества на тестовых корпусах и генеративная структура сделали данную модель одной из самых широко применяемых моделей анализа текстовых данных. Генеративная структура, при этом, позволила создать различные модификации, ориентированные на более узкий круг задач. В частности модификации, ориентированные на работу с короткими текстами и модификации, созданные для работы с мультиязычными данными.

Основные достоинства модели:

- Использование дополнительного распределения тем по документам делает данную модель более сложной относительно модели униграмм [33], в которой каждый документ задается одной темой.
- LDA является, в отличие от pLSI [34], генеративной моделью ввиду отсутствия дополнительно набора случайных переменных, задающих

модель pLSI. Данный факт позволяет LDA присваивать набор тем для документа, изначально не входящего в текстовый корпус.

Основные недостатки модели:

- Большая чувствительность к шуму
- Проблемы при работе с большими корпусами коротких текстов - в данном случае на вход данного алгоритма подается разреженная матрица в которой несмотря на потенциально большое число элементов, каждый отдельный элемент имеет небольшое число параметров.

1.2.2. Word Network Topic Model и Biterm Topic Model

Biterm Topic Model (BTM) - это алгоритм, который применяет процесс вывода, очень похожий на алгоритм LDA. Ключевым отличием этих алгоритмов является прямое использование шаблонов совпадения слов в BTM, т.е. в качестве базовой единицы берется не отдельное слово, а два слова, встречающиеся в общем контексте (концепт крайне схожий с использованием n-грамм). Это делает исходные данные более разнообразными и дает гораздо лучший результат при применении к коротким текстам, чаще всего встречающимся в социальных сетях [35]. Предыдущая работа с этой моделью показала свою эффективность по сравнению с базовой моделью LDA и другой модификацией процесса вывода - Word Network Topic Model. WNTM использует другой способ представления исходных данных, создавая полностью связанную сеть слов, позволяющую перейти от пространства документ-слово к пространству слово-слово [36]. Статья автора 2018 года [37] показала, что результаты, полученные с использованием BTM, более интерпретируемы с точки зрения оценки NPMI, однако частичная ручная оценка показала, что и модель WNTM, в определенных случаях, показывает качественный результат. Ключевым недостатком WNTM можно считать низкую скорость работы и выскоие требования по памяти, т.к. подготовительный этап включает в себя построение полной матрицы взаимной встречаемости слов.

Основные достоинства модели:

- Использование преобразования исходных данных к менее разряженных позволяет решить проблему коротких текстов

- Использование архитектуры схожей с LDA упрощает процесс имплементации

Основные недостатки модели:

- Сильное увеличение сложности как по памяти, так и по времени, ввиду преобразования данных в менее разреженную форму
- Склонность алгоритмов к переобучению
- Невозможность явного выражения тем в документах и как следствие - необходимость применения плохо обоснованных статистических преобразований

1.2.3. Bilingual Latent Dirichlet Allocation

BiLDA [38] это двуязычное расширение стандартной модели LDA, предназначенное для моделирования параллельных или, что еще более важно, сопоставимых тематически согласованных двуязычных коллекций документов. Примером такой коллекции документов является Википедия на 2 языках с парными статьями. BiLDA был независимо разработан несколькими исследователями. В отличие от LDA, где предполагается, что каждый документ имеет свое собственное специфичное для документа распределение по темам, процесс генерации для BiLDA предполагает, что каждая согласованная пара документов разделяет одинаковое распределение тем. Следовательно, модель предполагает, что у нас уже есть выравнивания документов в корпусе, то есть ссылки между парными документами на разных языках в двуязычном (или многоязычном) корпусе. BiLDA использует предполагаемое тематическое согласование на уровне связанных документов, вводя единую переменную, совместно используемую обоими документами. Данный подход генерализуется на случай когда исходные данные представлены больше, чем двумя языками, и позволяют получать тематические модели не так сильно зависимые от языка, однако требуют при этом специфичный формат данных для обучения.

Основные достоинства модели:

- Возможность кодировать мультиязычные наборы данных
- Использование проверенной архитектуры LDA

Основные недостатки модели:

- Необходимость соответствий текстов на разных языках
- Иные стандартные ограничения моделей на основе LDA

1.2.4. Нейросетевые подходы

В работах прошлых лет [39] были рассмотрены две основных нейросетевых модели для тематического моделирования, использующие подходы трансферного обучения для кодирования данных и представления тем.

Neural Variational Document Model

Neural Variational Document Model [40] заменяет стандартный подход к выражению тем из первоначальных данных. Стандартный подход модели LDA и ее различных улучшений, таких как BTM и WNTM заключается в моделировании базового набора тем с использованием вариационного вывода для вычисления параметров, описывающих вероятности для документа в коллекции. Результатом этого алгоритма является набор распределений: слов по темам и тем по документам для исходного набора данных. Модель NVDM, в отличие от подходов рассматриваемых ранее, вводит нейронную сеть для параметризации полиномиального распределения тем. Архитектура, лежащая в основе NVDM, эффективно использует преимущества рекуррентных сетей (RNNs) для моделирования последовательных текстовых данных. Подобная архитектура имеет преимущества по сравнению с традиционными типами нейронных сетей, такие как возможность обрабатывать входные данные любой длины (хотя в этом случае максимальная длина все еще ограничена мощностью параметрических моделей), возможность использовать информацию, полученную много шагов назад, и надежность с точки зрения потребления памяти в зависимости от длины ввода.

Embedded Topic Model

ETM [41] отличается от классических подходов, таких как LDA и его новых, более надежных, реализаций и модификаций, таких как BTM и NVDM тем, что в нем используются эмбеддинги (кодировки) для представления как слов, так и тем. Это позволяет модели выявлять сходства тем и слов без использования каких-либо внешних данных путем непосредственного моделирования их представлений. До открытия таких моделей для получения этой информации использовались внешние графовые данные о сходстве слов, такие как WordNet. ETM использует CBOW - тип языковой модели, которая, учитывая список окружающих слов, может выводить пропущенное слово в последовательности на основе совместных вероятностей вхождения. Авторами оригинальной статьи предлагается два основных варианта использования этой модели. Первый основан на обучении эмбеддингов слов как часть процесса вывода тем, одновременно получая эмбеддинги и для тем. Второй - на использовании внешних, заранее предобученных эмбеддингов, для получения результата с учетом уже существующего пространства кодировок. Последнее позволяет использовать слова, отсутствующие в обучающих данных, и может выполнять те же функции, что и графы сходства слов. Он улучшает лексическую емкость моделей за счет моделирования синонимов и слов, тесно связанных по значению, что особенно важно в случае социальных сетей из-за специфики и разреженности используемого языка. Еще одно преимущество этого подхода - надежность в отношении стоп-слов в словаре. Другие методы, такие как LDA, требуют тщательной предварительной обработки и в процессе вводят нежелательные переменные, такие как нижняя граница отсечки словаря. При этом кодировки обходят эту проблему, распознавая место стоп слов в пространствах эмбеддингов и помещая их в отдельную тему.

Основные достоинства нейросетевых подходов:

- Модели способны понимать и моделировать схожие слова
- Расширения возможных вариантов архитектуры - нейросетевые подходы более разнообразны по сравнению с статистическими и генеративными

Ключевые недостатки описанных нейросетевых моделей:

- Неспособность в полной мере использовать контекстные кодировки текстов
- Общие ограничения тематических моделей в части представления результатов - сложно интерпретируемы и требуют специфических представлений и визуализаций

1.3. Методы суммаризации

Существует два основных подхода к суммаризациям текстов: экстрактивная и абстрактная. Первый подход был распространен долгое время и наиболее сравним с традиционными методами на основе кластеризации, классификации и тематического моделирования. Второй подход стал предметом многих исследований начиная с 2015 года, однако приобрел особую популярность начиная с 2019 года, так как именно появление Transformer архитектуры позволило достигнуть качественных результатов для seq2seq моделей, которые и являются основой абстрактной суммаризации.

1.3.1. Экстрактивная суммаризация

Экстрактивные системы создают суммаризации путем определения (и последующего объединения) наиболее важных предложений в документе. Нейронные модели рассматривают экстрактивную суммаризацию как проблему классификации предложений: нейронный кодировщик создает представления предложений, а классификатор предсказывает, какие предложения следует выбрать в качестве суммаризаций. SUMMARUNNER [42] - один из первых нейронных подходов, использующих кодировщик на основе рекуррентных нейронных сетей. REFRESH [43] - это система, основанная на обучении с подкреплением, обученная путем глобальной оптимизации метрики ROUGE. В более поздних работах достигается более высокая производительность с более сложными модульными структурами. LATENT [44] рассматривает экстрактивное обобщение как проблему вывода скрытых переменных; вместо того, чтобы максимизировать вероятность суммирующих предложений, их скрытая модель напрямую максимизирует вероятность ручных суммаризаций. SUMO [45] использует понятие структурированного внимания для создания представления документа в виде дерева зависимостей с несколькими корнями при прогнозировании суммаризаций. NEUSUM [46] оценивает и отбирает предложения совместно и на момент выхода был одной из наиболее качественных моделей экстрактивной суммаризации.

1.3.2. Абстрактная суммаризация

Нейронные подходы к абстрактной суммаризации концептуализируют задачу как задачу получения одной последовательности слов из другой (seq2seq), где кодировщик отображает последовательность токенов в исходном документе $x = [x_1, ..., x_n]$ в последовательность непрерывных представлений $z = [z_1, ..., z_n]$, и декодер затем генерирует целевую сводку $y = [y_1, ..., y_m]$ токен за токеном авторегрессивным способом, таким образом моделируя условную вероятность: $p(y_1, ..., y_m|x_1, ..., x_n)$. Nallapati и др. (2016) стали одними из первых, применивших модель с использованием архитектуры нейронного кодировщика-декодера для суммаризации текстов [47]. В дальнейшем, данная модель была усовершенствована с помощью сети генераторов указателей (PTGEN) [48], которая позволяет копировать слова из исходного текста, и механизма покрытия (COV), который отслеживает слова, которые были суммированы. В методе Deep Communicating Agents (DCA) [49] предлагается система, в которой несколько агентов (кодировщиков) представляют документ вместе с иерархическим механизмом внимания (через агентов) для декодирования. Данная модель использует сквозное обучение с обучением с подкреплением. Paulus и др. (2018) также представляют глубокую модель с подкреплением (DRM) [50] для абстрактных суммаризаций, которая решает проблему охвата с помощью механизма внутреннего внимания, когда декодер обрабатывает ранее сгенерированные слова. Gehrmann и др. (2018) следуя восходящему подходу (BOTTOMUP), сначала определяют, какие фразы в исходном документе должны быть частью суммаризации, а затем, механизм копирования применяется только к заранее выбранным фразам во время декодирования [51]. Narayan и др. (2018) предлагают абстрактную модель, которая особенно хорошо подходит для экстремального случая - суммаризаций из одного предложения, основанного на сверточных нейронных сетях, дополнительно обусловленных распределением тем (TCONVS2S) [52].

1.3.3. Применение языковых моделей

Кроме названных ранее стандартных нейросетевых подходов, начиная с 2018 года предлагались различные варианты применения универ-

сальных языковых моделей как в задаче экстрактивных так и абстрактных суммаризаций. Одним из примеров может стать работа Yang Liu и Mirella Lapata в которой авторы рассматривают возможность применения модели BERT в задаче суммаризации текстовых данных [53].

Экстрактивная суммаризация BERT

Хотя BERT использовался для в решении целого спектра различных задач НЛП, с помощью методов дообучения и настройки модели, его применение в задачах суммаризации невозможно напрямую. Поскольку BERT обучается как замаскированная языковая модель, выходные векторы привязаны к токенам, а не предложениям, в то время как при экстрактивной суммаризации большинство моделей манипулируют представлениями на уровне предложений. Хотя эмбединги сегментации представляют разные предложения в BERT, они применимы только к входным парам предложений, в то время как при суммаризации возникает необходимость кодирования и манипулирования входными данными, относящимися к нескольким предложениям. На рис. 1.5. показана предлагаемая авторами оригинальной статьи архитектура BERT для суммаризаций (BERTSUM).

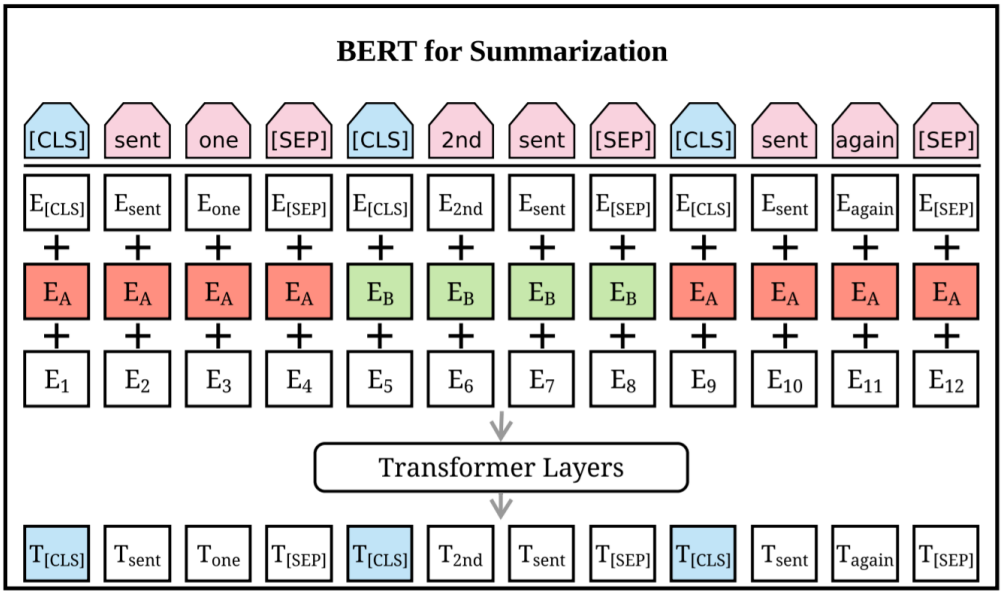

Рис. 1.5. Архитектура модификации BertSum

На данной схеме верхняя строка - входные предложения с разделителями. Следующий слой представляет собой сумму трех различных представлений слов (аналогично стандартной архитектуре BERT). Данная сум-

ма векторов используются в качестве входных кодировок для слоев двунаправленного преобразователя, генерируя контекстные векторы для каждого токена. BERTSUM расширяет BERT, вставляя несколько символов $[CLS]$ для изучения представлений предложений и используя эмбединги интервальной сегментации (показанные красным и зеленым цветом) для различия нескольких предложений.

Пусть d обозначает документ, содержащий $[sent_1, ..., sent_m]$, где $sent_i$ - это i-е предложение в документе. Экстрактивная суммаризация определяет такую задачу как задачу присвоения класса $y_i \in 0, 1$ каждому $sent_i$, показывающего включается ли объект в суммаризацию при $y_i = 1$ или не включается при $y_i = 0$. Предполагается, что собранные таким образом предложения представляют наиболее важные части исходного документа. С помощью BERTSUM вектор t_i, который является представлением i-го символа $[CLS]$ верхнего уровня, может использоваться в качестве представления для $sent_i$. Затем несколько слоев Transformer между предложениями складываются поверх выходных данных BERT, чтобы захватить функции уровня документа для извлечения суммаризации:

$$\tilde{h}^l = LN(h^{l-1} + MHAtt(h^{l-1}))$$

$$h^l = LN(\tilde{h}^l + FFN(\tilde{h}^l))$$

где $h^0 = PosEmb(T)$, T - выход векторов предложений, а функция $PosEmb(*)$ добавляет в кодировки синусоидную позиционную часть, показывая положение каждого предложения. Финальный слой данной сети представляет из себя следующий сигмоидный классификатор

$$\hat{y}_i = \sigma(W_0 h_i^L + b_0)$$

где h_i^L это векторное представление для предложения $sent_i$ из верхнего слоя Transformer модели.

Абстрактная суммаризация BERT

В отличии от экстрактивной суммаризации в данной задаче возможно использование стандартной архитектуры кодировщика-декодировщика.

Кодировщик - это предварительно обученная модель BERTSUM, а декодировщик - это 6-слойный преобразователь, инициализируемый случайным образом. Вполне возможно, что существует несоответствие между кодером и декодером, поскольку первый предварительно обучен, а второй должен обучаться с нуля. Это может сделать тюнинг модели нестабильным. Так например, кодировщик может переобучится, а декодировщик обучиться недостаточно, или наоборот. Чтобы обойти эту проблему авторы оригинальной статьи предлагают двухэтапный подход к тонкой надстройке (fine-tuning), при котором сначала настраивается кодировщик с помощью задачи экстрактивной суммаризации, а затем настраивается для задачи абстрактной суммаризации. Li и др. в своей работе показывают, как использование экстрактивных задач может повысить эффективность абстрактных суммаризаций [54]. Ввиду относительной простоты данного подхода, отсутствия изменений в основе архитектуры модели и общего улучшения результатов тюнинга модели, данный подход является стандартным при обучении данной модели.

1.4. Методы анализа тональности

Тональность можно определить как взгляд или мнение, содержащееся в некотором тексте. Sentiment Analysis при этом определяется как это процесс идентификации и категоризации мнений, выраженных в фрагменте текстового контента, в частности, для определения отношения автора к определенной теме, продукту или проблеме [55]. Sentiment Analysis, также известный как opinion mining, широко используется во многих областях, таких как анализ продуктов, услуг, социальных и политических проблем, для анализа поведения или мнения пользователей по определенным темам [56]. Целью анализа тональности является выявление положительных, отрицательных или нейтральных настроений в наборе текстов или документов [57].

1.4.1. Традиционные подходы

Традиционно данная задача решалась двумя основными подходами: словарным и с помощью методов машинного обучения [58].

Подходы, основанные на словарях, представляют собой статистические методы, использующие предварительно собранные словари тональности, содержащие различные слова и соотвествующую им полярность, для определение заданного слова как положительного или отрицательного. Stone и др. [59] впервые обозначили задачу анализа тональности с использованием метода словарей еще в 1966 году. Позже были предложены различные словарные наборы, такие как WordNet, WordNet-Affect, SenticNet, MPQA и SentiWordNet [60]. Эти подходы не требуют набора обучающих данных. Однако построение полных словарей для больших объемов неструктурированных данных, генерируемых пользователями, является сложной задачей.

Подходы на основе машинного обучению помогают решить проблему. Такие подходы основаны на алгоритмах классификации слов по соответствующим меткам тональности. Основное преимущество подходов на основе машинного обучения - их способность к репрезентативному обучению. Pang и др. [61] впервые применили эти методы для анализа тональности. Алгоритмам машинного обучения требуется обучающий набор дан-

ных, который помогает автоматизировать классификатор и тестовый набор данных, используемый для проверки работоспособности классификатора. Поэтому подходы машинного обучения предпочтительнее для анализа настроений из-за их способности работать с большими объемами данных по сравнению с подходами, основанными на словарях [62].

Несмотря на свою простоту и интуитивно понятную систему работы, традиционные алгоритмы имеют множество проблем, особенно заметных при работе с короткими текстами, полученными из социальных медиа. Именно поэтому современные модели все чаще создаются на основе различных архитектур глубокого обучения, в частности классификации на основе языковых моделей.

1.4.2. Подходы на основе нейросетей

За последние годы исследователями было предложено множество подходов на основе глубокого обучения, большинство из которых, так или иначе, были применены для анализа тональности. Среди таких подходов можно назвать предобученные сети без учителя (UPNs), сверточные сети (CNNs), рекуррентные сети (RNNs), рекурсивные сети (RvNNs) и описанные ранее трансформер сети. В разное время эти подходы показывали лучшие результаты, однако только с началом использования трансформер сетей и идей трансферного обучения, подходы на основе глубокого обучения практически полностью вытеснили традиционные методы. Новые подходы решили сложные проблемы, такие как адаптация предметной области, возможность работать с контекстом и моделировать долгосрочные зависимости.

1.5. Анализ коммерческих инструментов

1.5.1. IBM Watson

IBM предлагает набор инструментов, которые извлекают и классифицируют информацию в структурированных или неструктурированных текстовых данных. Среди инструментов данной системы можно выделить - IBM Watson Natural Language Understanding & Classifier, Watson Personality Insights и Watson Tone Analyzer.

IBM Watson Natural Language Understanding извлекает понятия, сущности, ключевые слова и категории. При используется для анализа тональностей, он не только сортирует текст на общие группы настроений - положительные, отрицательные и нейтральные - он также сортирует эти настроения по отдельным эмоциям, таким как смущение, грусть, уверенность и другие.

Классификатор естественного языка IBM Watson позволяет разработчикам извлекать значение из текста и использовать методы классификации, все это без необходимости быть экспертом в области машинного обучения или статистических алгоритмов. Дополнительно, разработчики могут создавать свои собственные модели, загружая свои данные и позволяя модели классифицировать тексты, извлекать аналитические данные и определять тенденции.

К числу основных достоинств системы можно отнести:

- Удобство работы, обширная документация
- Отсутвие необходимости в специалистах анализа данных даже для обучения моделей на собственных данных
- Широкий спектр решаемых задач – от базовой обработки, до анализа эмоций и суммаризаций
- Гибкая масштабизация (система лицензии Pay as you go)

Основными недостатками продукта являются:

- Относительно высокая стоимость. В примерах документации приводится расчет стоимости анализа тональности для 20 тыс. твитов, которая составляет 60 USD. Реальные наборы данных при этом часто

насчитывают миллионы сообщений и как следствие, стоимость их единоразового анализа составит десятки тысяч долларов.

- Использование устаревших способов анализа тональности - n-граммные кодировки и классификация с помощью svm (support vector machine)
- Использование устаревших нейросетевых методов суммаризации - Pointer-Generator Networks [63]

1.5.2. MeaningCloud

MeaningCloud предлагает облачные API-интерфейсы и графические интерфейсы для выполнения задач анализа текста. Данная система включает в себя автоматическую суммаризацию, категоризацию, анализ структуры документа, анализ тональности, классификацию текста, кластеризацию текста и извлечение тем (извлечение тем при этом не соотносится с стандартным определением тематического моделирования и выполняет роль более близкую к задаче распознавания именных сущностей - named entity recognition).

API комплекса позволяет добавлять собственные словари, чтобы помочь моделям сосредоточиться на аспектах / функциях конкретного продукта, корректировать технические тексты без необходимости вручную игнорировать более конкретные термины, которые не включены в базовые словари, и извлекать упоминания сущностей и концепций в тексте, присваивая им значение исходя из собственной онтологии.

Дополнительно MeaningCloud предлагает вертикальные пакеты, ресурсы, адаптированные для конкретного приложения или отрасли. Например, вертикальный пакет «Голос клиента» включает ресурсы или компоненты, специально предназначенные для банковских, страховых и телекоммуникационных отраслей.

Основными достоинствами являются:

- Удобство работы, интеграция с офисными приложениями: Excel, Google Spreadsheets
- Поддержка большого числа языков, с постоянными обновлениями

- Модель распространения SaaS и, как следствие, гибкие возможности лицензирования – бесплатное использование для 20 тыс. запросов в месяц и до 1000 USD для 4200 тыс. ежемесячных запросов.

К ключевым недостаткам можно отнести:

- Использование устаревших технологий на основе гибридных методов машинного обучения и правил
- Низкая скорость обработки данных - 2 запроса в секунду для бесплатной лицензии и до 15 запросов в секунду для бизнес-лицензии

Глава 2

Система анализа пользовательских дискуссий

В рамках данной работы была разработана архитектура программного комплекса, включающего сбор данных, их предварительную обработку, агрегацию, кодирование и реализацию различных методов кластеризации, тематического моделирования, суммаризации. Структура была выполнена в виде библиотеки для Python и включает в себя реализацию всех компонентов, включая визуализацию и представление результатов.

2.1. Архитектура комплекса

На рис. 2.1. представлена архитектура программного комплекса. Блоки данной схемы представляют компоненты системы и включают в себя: тип операции (сбор, обработка, агрегация и т.д.), входные параметры (типы и гиперпарметры моделей и т.д.) и выходной результат отдельных этапов (кодировки, графики, выборки и т.д.). Пунктиром обозначены опциональные этапы. В случае связи сбора данных и их предобработки, выполнение второго этапа опционально т.к. не все методы кодировки требуют тщательной подготовки данных с приведениме к начальной форме (стемингом или лемматизацией) и удаления стоп слов (такими словарными библиотеками как Python stop-words).

Результаты кластеризации опционально возможно использовать для определения числа тем в задаче тематического моделирования. Все алгоритмы центральной части схемы (Тематическое моделирование, Абстрактная суммаризация, Кластеризация и Классификация тональности) возможно использовать совместно для получения наиболее информативных результатов анализа. Дальнейшие разделы демонстрируют последовательный анализ для двух наборов данных.

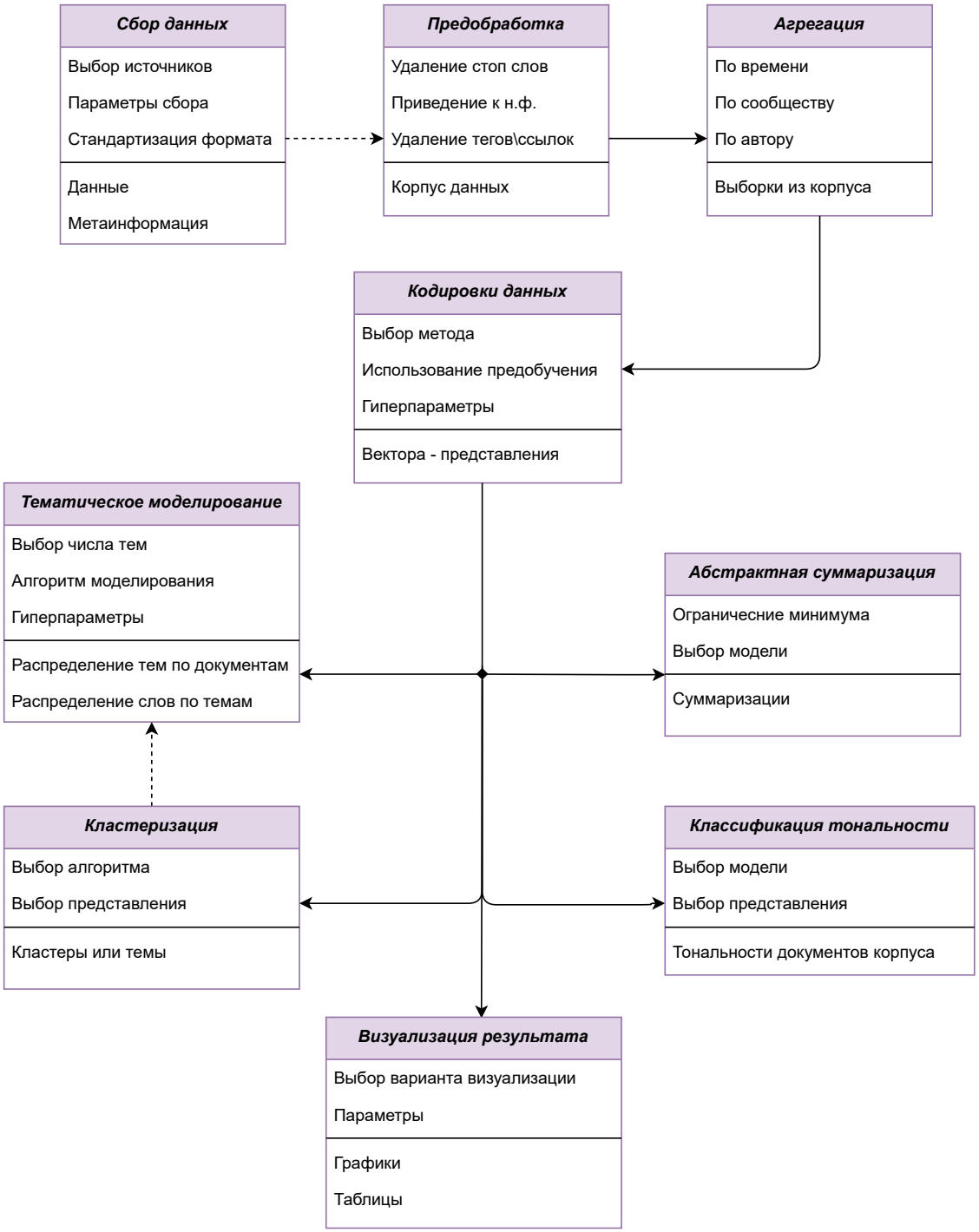

Рис. 2.1. Архитектура программного комплекса

2.2. Данные - сбор и анализ

Для проведения анализа было собрано два набора данных, отличающихся по многим параметрам. Первый набор был собран из социальной сети Twitter с помощью собственной системы сбора [64]. Рассматриваемый набор основан на сообщениях, касающихся террористического акта в офисе редакции Charlie Hebdo (Франция) в январе 2015 года. Сбор данных производится по заранее выявленным ключевым словам и тегам, характеризующим рассматриваемую дискуссию. Заранее задается также и временной промежуток в течении которого рассматриваются записи, для ограничения их числа и агрегации сообщений относящихся непосредственно к обсуждению пользователей на заданную тему.

Мультиязычность является одной из ключевых особенностей данного набора данных. Дискуссия, в основном велась на английском и французском языках, однако насчитывает множество других языков. На рис. 2.2. приводится график соотношения языков в корпусе, полученный с помощью предобученной системы на основе fasttext [65].

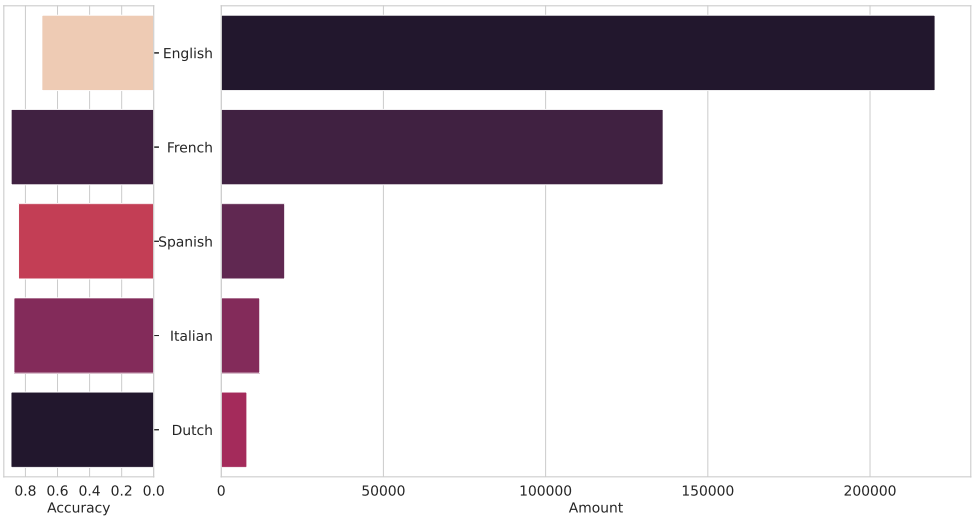

Рис. 2.2. Распределение языков в первом наборе данных

Исходный набор данных составляет 420080 постов, на данном графике отображены средние значения точности определения каждого языка и число отнесенных к ним постов. Ввиду достаточно высокой точности определения и наибольшего числа постов в дальнейшем анализе будут рассмат-

риваться сообщения на английском и французском языках.

На рис 2.3. показано изменение числа постов на английском и французском языках с течением времени.

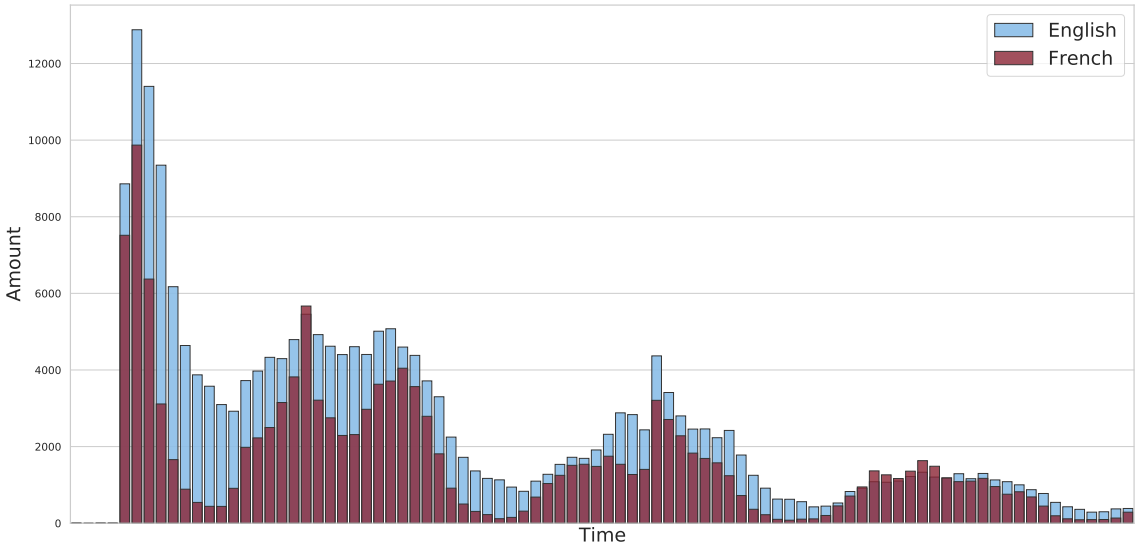

Рис. 2.3. Изменения числа постов с течением времени

Из данного графика видна значимость рассматриваемых языков, т.к. число сообщений на французском, в определенные часы, превышает число постов на английском, при этом разные языки имеют схожую динамику.

На рис 2.4. представлен график плотностного распределения размера постов на французском и английском.

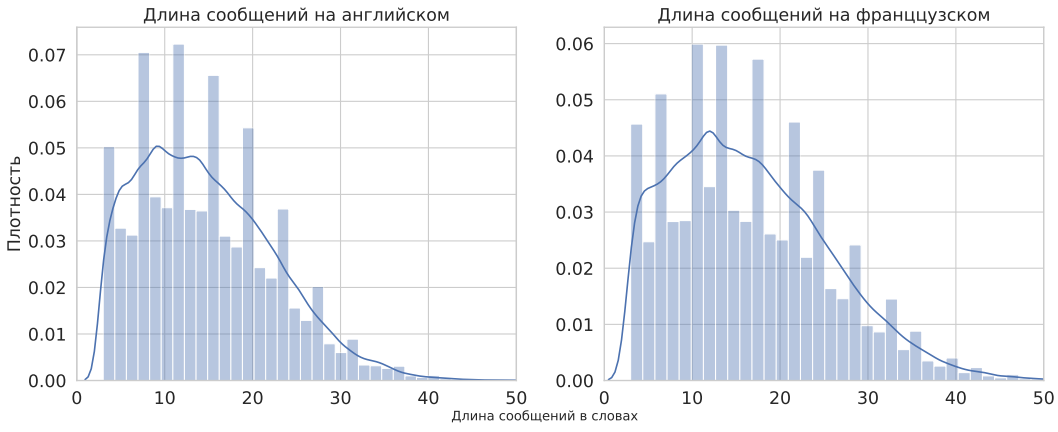

Рис. 2.4. Плотностное распределение длины постов

Из данных распределений видно, что для постов и на английском, и на французском сообщения крайне короткие и имеют среднюю длину 12 слов, с максимальными размерами не превышающими 50 слов.

Второй набор данных был собран из социальной сети Reddit, представляющей собой платформу для участия в сообществах. Данные собирались по двум сообществам с полярными мнениями: r/republican и r/democrats. Выбор данных страниц обусловлен ожидаемой разностью их участников, схожими размерами как по числу участников, так и по уровню взаимодействия с контентом. Сбор производился по популярным постам, для каждого из которых дополнительно собирались 30% популярных комментариев.

Данный набор сильно отличается от корпуса данных из Twitter по нескольким ключевым параметрам:

- Одноязычные данные - т.к. платформа в основе своей англоязычная, а обсуждения в рассматриваемых сообществах касаются, в первую очередь, американской политической системы
- Меньшие объемы данных - не является признаком платформы в целом, относится к рассматриваемому примеру, т.к. при поиске сообществ основной целью было найти схожие по количеству участников и уровню взаимодействия. Итоговый объем данных – 1996 постов и 47141 вложенных в них комментариев.
- Намного более длинные посты и комментарии пользователей

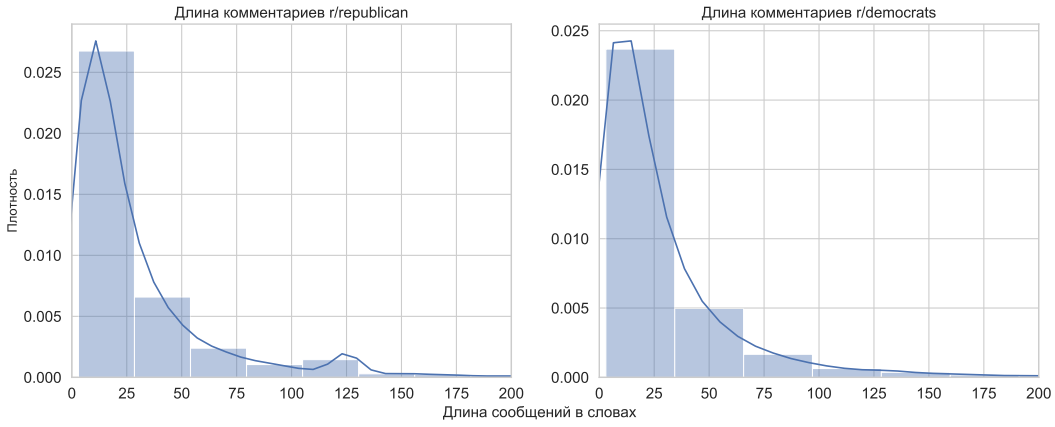

Рис. 2.5. Плотностное распределение длины постов

На рис. 2.5. показаны распределения размеров постов и комментариев из которых видно, что средняя и максимальная длина постов превышает значения Twitter данных в 4 раза.

Рис. 2.6. показывает рост числа популярных постов и распределение рейтингов в двух сообществах.

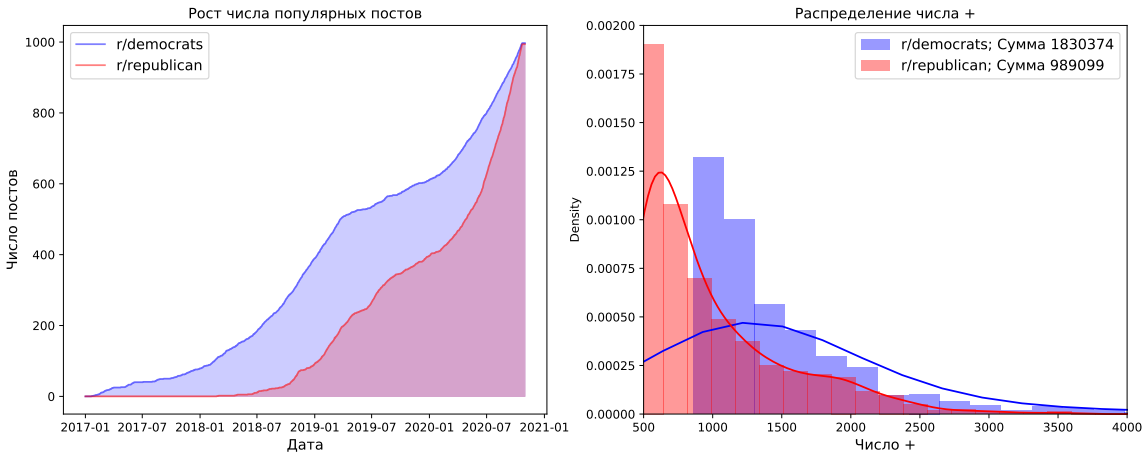

Рис. 2.6. Сравнение популярности

На данных сравнениях видно, что сообщества близки как по количеству популярных постов, так и по распределению рейтингов (последнее с перевесом в сторону r/democrats, однако на дальнейшие тесты, данный фактор не влияет).

Полученные наборы данных позволяют оценивать качество различных алгоритмов в различных условиях, при этом придерживаясь темы анализа пользовательских дискуссий в социальных сетях.

2.3. Кластерный анализ

Данный раздел во многом базируется на идее метода Top2Vec [66]. Основная идея метода заключается в использовании языковой модели для кодирования данных, с дальнейшим уменьшением размерности, применением алгоритма кластеризации и получением тем статистическими методами.

Первым этапом данного метода является кодирование данных и уменьшение размерности. Для данной задачи первичные тесты проводились на наборе данных, полученном из Twitter, ввиду отсутствия ограничений на способ представления и как следствие - возможности в полной мере использовать мультиязычные кодировщики. Авторы оригинальной статьи Top2Vec используют модель word2vec, однако для получения качественных результатов на мультиязычных данных, оправдано использование мультиязычной модели Universal Sentence Encoder, ранее описанной в разделе 1.1.7. Используя Universal Sentence Encoder Cross-lingual (XLING) [67] с предобученными параметрами для английского и французского языков позволяет кодировать исходные данные в 512-мерные векторы.

Для сокращения времени проведения тестов, изначальные кодировки сжимаются алгоритмом UMAP [68] до 50-мерных векторов.

Следующий этап - кластеризация на основе полученных векторных представлений. В данном случае рационально применение алгоритма HDBSCAN [69]. Данный алгоритм является модификацией широко распространенной модели DBSCAN - модифицирует базовый алгоритм, позволяя находить кластеры с различными плотностями, не требуя при этом ручной оценки порога объединения кластеров. Такой подход позволяет не только получать кластерные разбиения высокого качества, но и автоматически определять шумовые документы, не относящиеся ни к одному кластеру, и число кластеров. Визуальное представление документов может быть получено путем уменьшения размерности до 2 измерений.

Чтобы получить репрезентации получившихся 698-и кластеров необходимо получить их разбиения по словам. В простейшем случае данные разбиения можно получить с помощью алгоритма TF-IDF, описанного в разделе 1.1.2. Для применения данного подхода объединяются предложения, относящиеся к одному кластеру. Применение алгоритма TF-IDF на данном представлении позволяет получать наиболее значимые слова для

каждого кластера.

В таблице 2.1. представлены слова, определяющие некоторые популярные кластеры.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
satire	identite	victims	israel	shooting
freedom	suspects	terror	palestine	gunmen
satirical	police	paris	gaza	suspects
free	localises	attack	hamas	photos
speech	identifies	remember	attacks	connue

Таблица 2.1. Распределение слов по кластерам

Аналогичные тесты для объединенного набора данных Reddit показывают наличие 250 кластеров. На рис. 2.7. показано распределение кластеров по документам, а в таблице 2.2. представлены наиболее важные слова из основных нешумовых кластеров.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
twitter	nfl	hollywood	propaganda	black
facebook	sports	celebrities	machine	white
social	players	actors	misinformation	racist
media	anthem	fame	algorithms	blm
zuckerberg	knee	like	conspiracy	police

Таблица 2.2. Распределение слов по кластерам

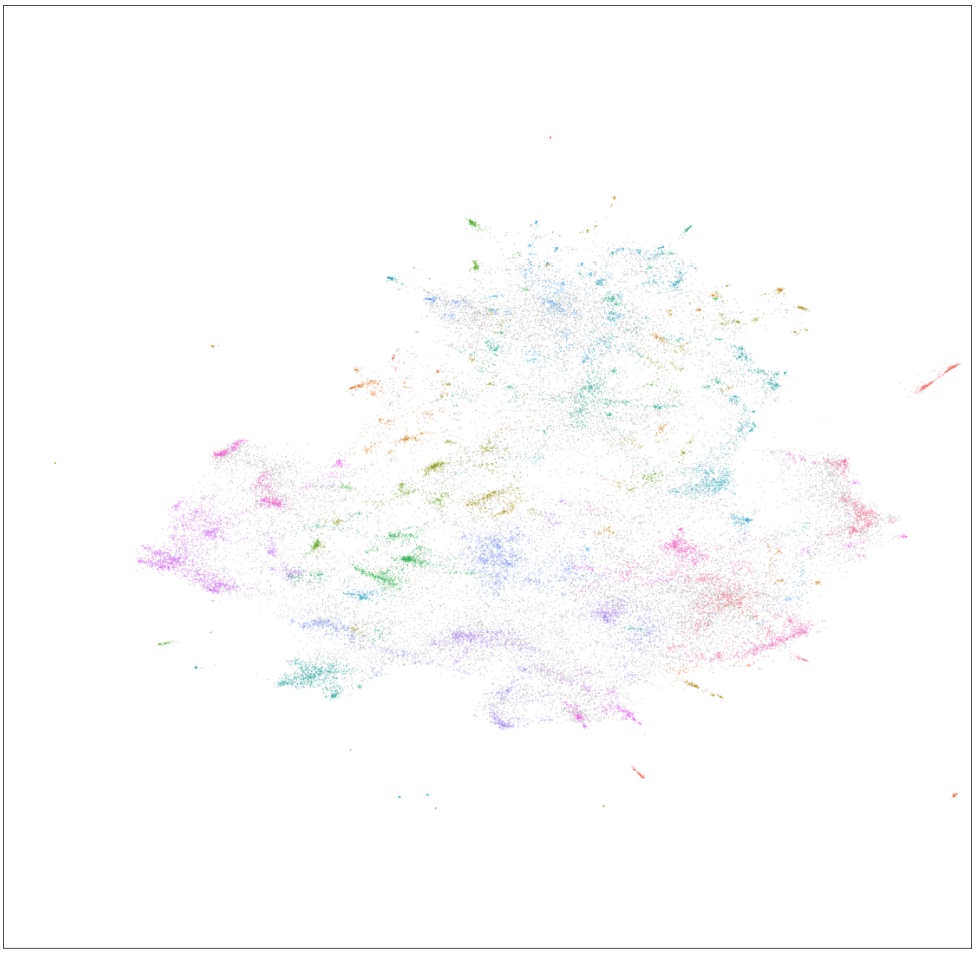

Рис. 2.7. Распределение документов по кластерам

2.4. Тематическое моделирование

Тематическое моделирование ставит своей задачей получение двух основных распределений: тем по документам и слов по темам. Наиболее распространенными подходами решения данной задачи являются генеративные методы, основанные на архитектуре LDA. Прошлые тесты [39] показали эффективность модели Embedded Topic Model по сравнению с базовым методом LDA и его модификациями, направленными на обработку коротких текстов. Высокие показатели эффективности связаны прежде всего с использованием архитектуры кодирования, позволяющей эффективно

кодировать шумные данные. В таком случае, как и в эксперименте с использованием USE, шумовые слова кодируются отдельной темой и почти не содержатся в других темах.

В данной работе использовалась модификация репозитория авторов статьи, с дополнительными функциями загрузки данных и получения результирующих матриц. В качестве числа тем были взяты значения полученные HDBSCAN в предыдущих тестах.

В таблицах 2.3. и 2.4. показаны распределения слов для некторох тем из объединенного набора Reddit и английской части данных, полученных из twitter.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
vote	election	news	virus	people
elected	gov	cnn	pandemic	white
just	2020	fox	coronavirus	racist
party	registration	people	algorithms	black
election	primary	media	just	political

Таблица 2.3. Распределение слов по темам для данных Reddit

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
religion	speech	respond	hostages	unite
islam	freedom	massacre	police	solidarity
muslim	free	cartoonists	hostage	victims
people	expression	world	hope	cartoonists
god	right	victims	pray	attack

Таблица 2.4. Распределение слов по темам для данных Twitter

2.5. Суммаризация дискуссий внутри отдельных постов

Часто в различных социальных медиа посты являются изображениями, ссылками на новостные ресурсы, видео и так далее. В таком случае есть несколько путей применения алгоритмов анализа данных. Первый - более сложный, заключается в переходе по ссылкам, сборе информации из оригинальных источников, выделении текстовой информации из картинок и так далее. Второй способ заключается в анализе комментариев к основной записи. Такой подход позволяет сокращать сложность обработки, так как нет необходимости прибегать к дополнительному этапу сбора дополнительной информации и алгоритмам выделения текстовых данных. Дополнительно, анализ комментариев позволяет выявлять мнения многих участников по теме, обсуждаемой в сообщении.

Данный анализ проводился для набора данных reddit, т.к. именно для этого набора производился сбор и постов, и комментариев (для Twitter сбор выполнялся по ключевым словам и тегам, и не содержит информации о структуре постов и комментариев).

Для создания суммаризаций комментариев использовалась модель Longformer, ввиду ее эффективности при суммаризации больших объемов данных. Самые популярные записи насчитывают тысячи комментариев и применение стандартных моделей, таких как T5 и BERTsum, возможно, но накладывает серьезные требования на оборудование (требуются большие объемы видеопамяти) и накладывает ограничения на размер получаемых суммаризаций. Модель Longformer, при этом, позволяет захватывать большую часть информации из большого числа комментариев, получая при этом качественные суммаризации.

Дополнительно, для данной задачи производится совместное с суммаризацией применение модели анализа тональности из набора Flair, предобученной для работы с английским текстом. Результаты анализа тональности комментариев к популярным постам двух сообществ приведены на рис. 2.8.

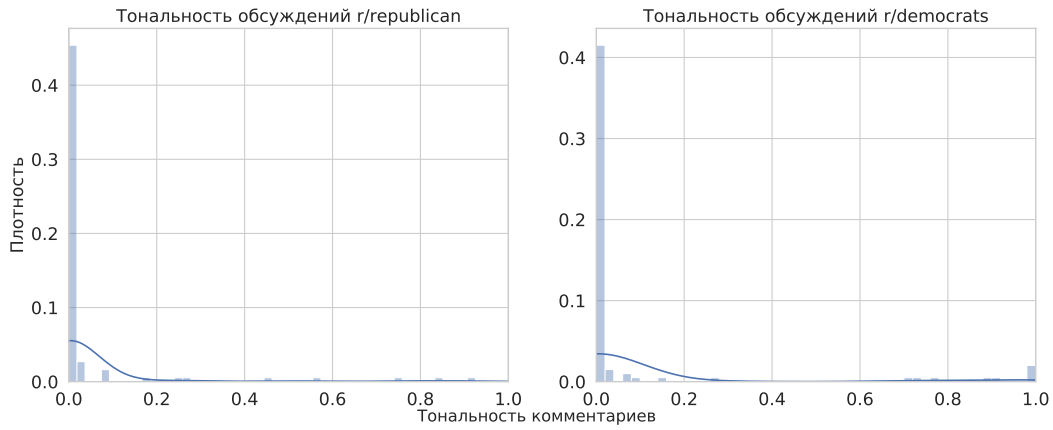

Рис. 2.8. Плотностное распределение тональности комментариев

Из данных распределений видно, что большинство комментариев, особенно для сообщества rbackslashchar republican являются негативными. Это объясняется главным образом популярностью обсуждений, касающихся новостных статей и событий, имеющих явно негативный окрас. Дополнительно можно отметить поляризованную оценку тональности, получаемую с помощью flair - большинство комментариев оценивается явно позитивно или негативно, т.е. оценивается значениями близкими к 0 или 1, с редкими промежуточными значениями.

На рис. 2.9. показана табличная визуализация суммаризаций комментариев к популярным постам и соответствующих им значениям тональности. Данное представление позволяет просматривать общий смысл сотен различных комментариев, анализируя одно, общее для них представление. Дополнительная разметка тональности позволяет быстро оценивать общий настрой комментариев, упрощает поиск по итоговой таблице.

Можно отметить высокое качество полученных суммаризаций. Они не только составляют связные предложения и идеи, но и соответствуют обсуждаемым темам во всех рассмотренных примерах. Оценка тональности flair, в свою очередь, требует значительной доработки. Так например седьмая суммаризация для rbackslashchar democrats содержит явный сарказм, но модель считает предложение позитивным.

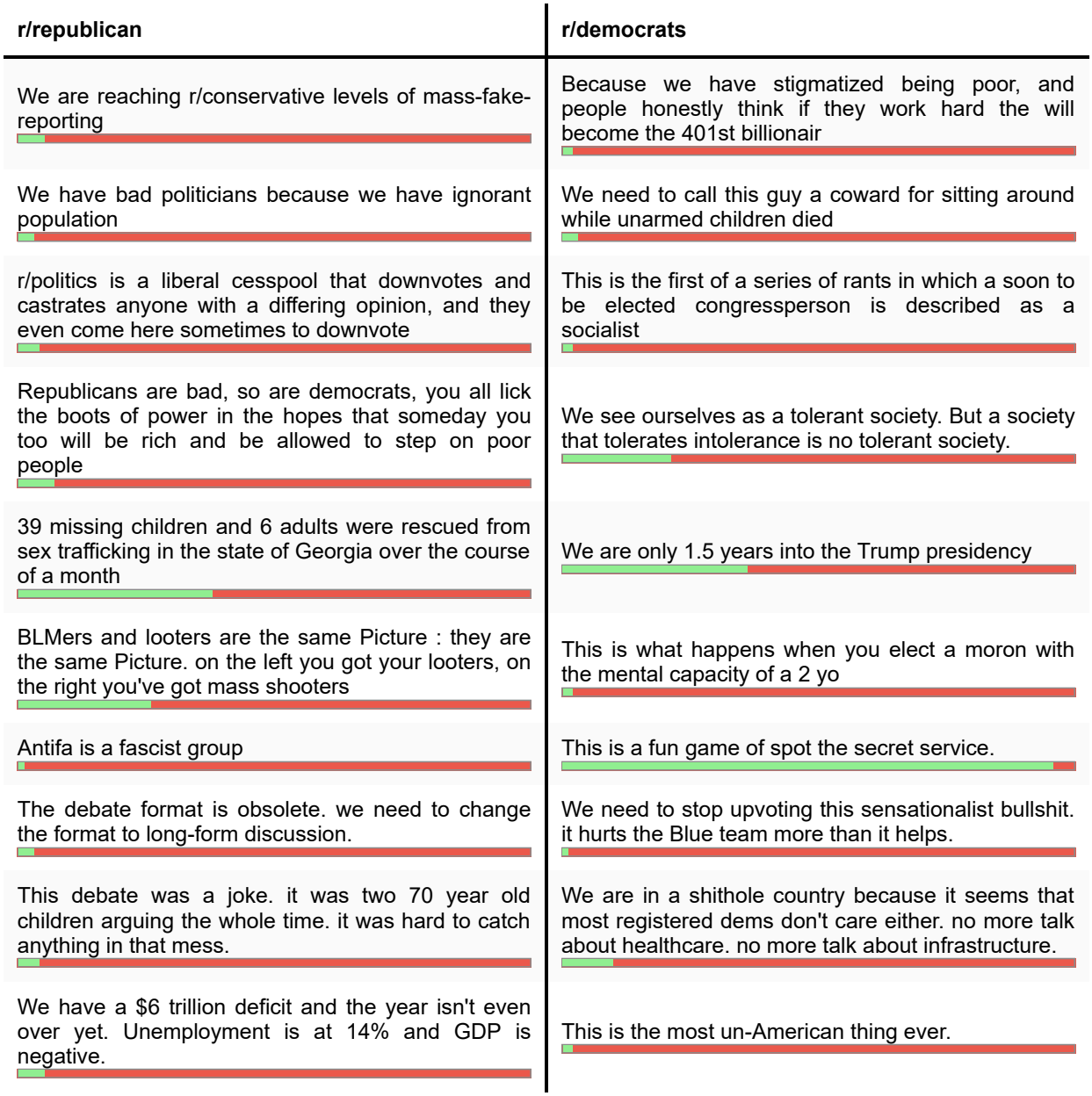

Рис. 2.9. Результаты анализа дискуссий в комментариях

2.6. Суммаризация и тональность временных рядов

В случае анализа дискуссий, касающихся некоторого события, особую важность имеет обработка сообщений за определенные временные промежутки. Ранее на рис. 2.3. приводился график изменения числа сообщений с течением времени из которого видно, что дискуссия протекает неравномерно.

Для проведения анализа дискуссии в Twitter данные были разбиты на пиковые промежутки, с последующим разделением по тональностям с помощью предобученной модели BERT. Выборки разбиваются на положи-

тельные при значении тональности $n \geqslant 0.6$, нейтральные при значениях $0.6 > n > 0.4$ и отрицательные при $n \leqslant 0.4$. Итоговые выборки суммируются моделью T5. На рис. 2.10. представлены результаты анализа временных распределений.

Начало промежутка	**Суммаризация**	**Соотношение тональностей**
2015-01-07 08:51	Condemn violence against media. Our thoughts go out to the families of the slain victims and their families in paris.	39%
	Sure the pin is stronger than weapons but not in Charlie way But they cannot kill us all. Sometimes a picture says it better than any writer could.	17%
	A terrible day for all cartoonists is a terrible day for freedom of speech. A sad day for freedom of speech, for creatives, and for the world .	43%
2015-01-07 12:27	r.i.p. is a cartoonist who reacted to the attack on freedom of speech. He says it's an act of cowardness and freedom of speech.	40%
	A terrible day for all cartoonists	20%
	A sad day for all cartoonists in the u.s. is when people are silenced for satirical cartoons. Tyranny, home and abroad, and love is always stronger than fear and hate.	39%
2015-01-07 12:30	Freedom of speech is our greatest weapon against tyranny, home and abroad.	46%
	As a Muslim, I refuse to apologize for this terrorist act.	17%
	A terrible day for all french cartoonists. Just awful.	37%
2015-01-07 12:33	Freedom of expression is our greatest weapon against tyranny, home and abroad. A pen is mightier than the sword.	46%
	A terrible day for all cartoonists, I demand justice for the victims.	19%
	Freedom of expression is inherently anti-oppression, whether religious or political.	34%

Рис. 2.10. Результаты анализа временных распределений

В данной таблице первый столбец указывает время начала рассматриваемого промежутка. Второй показывает три различных суммаризации: для положительного, нейтрального и негативного контента. Последний показывает распределение тональностей в данном временном промежутке.

Заключение

Результаты работы

В рамках данной работы был проведен сравнительный анализ различных моделей суммаризации текстовых данных и способов их совместного применения с методами определения тональности. Был проведен тщательный анализ современных подходов к решению задач кодирования текстовых данных, суммаризации, тематического моделирования и анализа тональности. Дополнительно был проведен анализ существующих коммерческих инструментов, показавший, что методы, используемые сервисами, часто являются устаревшими - основанными на эвристических подходах, подходах с использованием машинного обучения. При этом анализ литературы показывает, что наиболее качественные результаты для решения всех рассматриваемых задач достигаются с использованием нейросетевых подходов, а именно - с использованием архитектуры Transformer и подхода трансферного обучения.

Для решения практических задач была разработана архитектура, затрагивающая все основные этапы анализа данных, начиная от сбора и предобработки, заканчивая визуализацией полученных результатов. Описывается процесс сбора и первичного анализа двух наборов данных, полученных из разных соцсетей. Проводится ряд практических тестов, показывающих примеры получения значимых репрезентаций из исходных корпусов.

В качестве основных выводов можно отметить:

- Различные данные требуют подбора моделей, лучше приспособленных к работе с теми или иными специфическими ограничениями. Примерами таких ограничений могут быть: мультиязычность исходных данных, большой размер корпусов, использование коротких текстов.

- Методы кластерного анализа, при правильном их применении совместно с современными методами кодирования данных, позволяют получать качественные представления документов. Использование дополнительных статистических приемов позволяет получать словар-

ные репрезентации кластеров, приводя задачу к виду, схожему с задачей тематического моделирования. Модели имеют ряд ключевых преимуществ перед методами тематического моделирования: гибкая архитектура (возможность использования различных кодировок, включая мультиязычные), возможность автоматического определения числа кластеров плотностными алгоритмами.

- Классические и современные методы тематического моделирования имеют перед методами кластеризации ряд преимуществ, основными из которых являются прочная теоретическая основа и возможность кодирования документов одновременно несколькими темами. При этом преимущество нескольких тем на документ часто оказывается невостребованным в задаче анализа текстов, полученных из соцсетей. В таких случаях достаточным оказывается использование одной (самой популярной) темы на сообщение, ввиду короткого размера текстов.

- Подходы абстрактной суммаризации позволяют получать качественно иные репрезентации текстовых данных. Обобщения генерируемые такими методами способны качественно передавать смысл большого числа постов и комментариев в сжатой форме. В дальнейшем такие репрезентации могут быть использованы специалистами профильных областей (социологами, журналистами, политологами и маркетологами).

- Методы анализа тональности могут использоваться как на этапе агрегации данных (с дальнейшим анализом негативных или позитивных записей), так и на этапе финализации для получения дополнительных визуальных представлений.

Исследования отдельных тем данной работы были выполнены при поддержке гранта РНФ "Кривое зеркало конфликта: роль сетевых дискуссий в репрезентации и динамике этнополитических конфликтов в России и за рубежом"(грант 16-18-10125-Р). В рамках гранта были написаны три статьи, предметом которых стало исследование методов тематического моделирования и кластеризации данных конфликтных дискуссий. Также в рамках гранта был зарегистрирован патент "Программа для автоматического обнаружения скрытых тем в пользовательских дискуссиях"№ 2020662702.

Перспективы развития

Несмотря на полноту проведенного анализа, работа по данной теме может быть продолжена. В качестве основных направлений развития проекта можно выделить:

- Улучшение методов классификации тональности - использование алгоритмов и моделей способных показывать различные эмоциональные оттенки, а также способные детектировать сарказм
- Улучшение мультиязычной обработки данных - предобучение моделей, способных качественно обрабатывать одновременно различные языки
- Дальнейшая модификация Transformer подходов с помощью идей модели Longformer, для качественной и эффективной обработки больших объемов данных
- Более полное использование метаинформации при создании моделей (рейтинги записей, временные отметки комментариев, структура ответов к постам и комментариям)

Список литературы

1. Tankovska H. Number of global social network users 2017-2025. — 2021.
2. Aggarwal C., Zhai C. A survey of text clustering algorithms. // Mining text data. — 2012. — с. 77—128.
3. A comprehensive survey on transfer learning / F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He // Proceedings of the IEEE. — 2020. — т. 109, № 1. — с. 43—76.
4. Leopold E., Kindermann J. Text categorization with support vector machines. How to represent texts in input space? // Machine Learning. — 2002. — с. 423—444.
5. Sebastiani F. Machine learning in automated text categorization. // ACM computing surveys. — 2002. — с. 1—47.
6. Efficient Estimation of Word Representations in Vector Space / T. Mikolov, K. Chen, G. Corrado, J. Dean // arXiv e-prints. — 2013. — arXiv—1301.
7. Enriching word vectors with subword information / P. Bojanowski, E. Grave, A. Joulin, T. Mikolov // Transactions of the Association for Computational Linguistics. — 2017. — т. 5. — с. 135—146.
8. Deep Contextualized Word Representations / M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer // Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). — 2018. — с. 2227—2237.
9. Character-level language modeling with deeper self-attention / R. Al-Rfou, D. Choe, N. Constant, M. Guo, L. Jones // Proceedings of the AAAI Conference on Artificial Intelligence. т. 33. — 2019. — с. 3159—3166.
10. Jing K., Xu J. A survey on neural network language models. — 2019.

11. Learning phrase representations using RNN encoder-decoder for statistical machine translation / K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio // arXiv preprint arXiv:1406.1078. — 2014.

12. Cheng J., Dong L., Lapata M. Long short-term memory-networks for machine reading // arXiv preprint arXiv:1601.06733. — 2016.

13. Attention is all you need / A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin // arXiv preprint arXiv:1706.03762. — 2017.

14. Universal Sentence Encoder for English / D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar [и др.] // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. — 2018. — с. 169—174.

15. Deep unordered composition rivals syntactic methods for text classification / M. Iyyer, V. Manjunatha, J. Boyd-Graber, H. Daumé III // Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers). — 2015. — с. 1681—1691.

16. Kenton J. D. M.-W. C., Toutanova L. K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // Universal Language Model Fine-tuning for Text Classification. — 2018. — с. 278.

17. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer / C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu // Journal of Machine Learning Research. — 2020. — т. 21. — с. 1—67.

18. The Natural Language Decathlon: Multitask Learning as Question Answering / B. McCann, N. Shirish Keskar, C. Xiong, R. Socher // arXiv e-prints. — 2018. — arXiv—1806.

19. Language models are unsupervised multitask learners / A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever // OpenAI blog. — 2019. — т. 1, № 8. — с. 9.

20. Ba J. L., Kiros J. R., Hinton G. E. Layer Normalization // stat. — 2016. — т. 1050. — с. 21.

21. Deep residual learning for image recognition / K. He, X. Zhang, S. Ren, J. Sun // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2016. — с. 770—778.

22. Dropout: a simple way to prevent neural networks from overfitting / N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov // The journal of machine learning research. — 2014. — т. 15, № 1. — с. 1929—1958.

23. Buck C., Heafield K., Van Ooyen B. N-gram Counts and Language Models from the Common Crawl. // LREC. т. 2. — Citeseer. 2014. — с. 4.

24. Trinh T. H., Le Q. V. A Simple Method for Commonsense Reasoning // arXiv e-prints. — 2018. — arXiv—1806.

25. Dirt cheap web-scale parallel text from the common crawl / J. Smith, H. Saint-Amand, M. Plamadă, P. Koehn, C. Callison-Burch, A. Lopez // Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). — 2013. — с. 1374—1383.

26. Memory-Efficient Adaptive Optimization for Large-Scale Learning / R. Anil, V. Gupta, T. Koren, Y. Singer. — 2019.

27. SuperGLUE: A stickier benchmark for general-purpose language understanding systems / A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, S. R. Bowman // Advances in Neural Information Processing Systems. — 2019. — т. 32.

28. Beltagy I., Peters M. E., Cohan A. Longformer: The Long-Document Transformer // arXiv e-prints. — 2020. — arXiv—2004.

29. Revealing the Dark Secrets of BERT / O. Kovaleva, A. Romanov, A. Rogers, A. Rumshisky // arXiv e-prints. — 2019. — arXiv—1908.

30. Pay Less Attention with Lightweight and Dynamic Convolutions / F. Wu, A. Fan, A. Baevski, Y. N. Dauphin, M. Auli // arXiv e-prints. — 2019. — arXiv—1901.

31. WaveNet: A Generative Model for Raw Audio / A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu // arXiv e-prints. — 2016. — arXiv—1609.

32. Blei D. M., Ng A. Y., Jordan M. I. Latent dirichlet allocation // the Journal of machine Learning research. — 2003. — т. 3. — с. 993—1022.

33. Text classification from labeled and unlabeled documents using EM / K. Nigam, A. K. McCallum, S. Thrun, T. Mitchell // Machine learning. — 2000. — т. 39, № 2. — с. 103—134.

34. Hofmann T. Probabilistic latent semantic analysis // arXiv preprint arXiv:1301.6705. — 2013.

35. BTM: Topic Modeling over Short Texts / C. Xueqi, Y. Xiaohui, L. Yanyan, G. Jiafeng // IEEE Transactions on Knowledge and Data Engineering. — 2014. — с. 2928—2941.

36. Yuan Z., Jichang Z., Ke X. Word Network Topic Model: A Simple but General Solution for Short and Imbalanced Texts. — 2014.

37. Blekanov I., Tarasov N., Maksimov A. Topic modeling of conflict ad hoc discussions in social networks. // ACM International Conference Proceeding Series. — 2018. — с. 122—126.

38. Polylingual topic models / D. Mimno, H. Wallach, J. Naradowsky, D. A. Smith, A. McCallum // Proceedings of the 2009 conference on empirical methods in natural language processing. — 2009. — с. 880—889.

39. Blekanov I., Tarasov N., Maksimov A. Topic Models with Neural Variational Inference for Discussion Analysis in Social Networks. //. — 2020.

40. Miao Y., Yu L., Blunsom P. Neural variational inference for text processing // International conference on machine learning. — PMLR. 2016. — с. 1727—1736.

41. Dieng A. B., Ruiz F. J., Blei D. M. Topic modeling in embedding spaces // Transactions of the Association for Computational Linguistics. — 2020. — т. 8. — с. 439—453.

42. Nallapati R., Zhai F., Zhou B. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents // Proceedings of the AAAI Conference on Artificial Intelligence. т. 31. — 2017.

43. Narayan S., Cohen S. B., Lapata M. Ranking Sentences for Extractive Summarization with Reinforcement Learning // Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). — 2018. — с. 1747—1759.

44. Neural Latent Extractive Document Summarization / X. Zhang, M. Lapata, F. Wei, M. Zhou // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. — 2018. — с. 779—784.

45. Liu Y., Titov I., Lapata M. Single document summarization as tree induction // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). — 2019. — с. 1745—1755.

46. Neural Document Summarization by Jointly Learning to Score and Select Sentences / Q. Zhou, N. Yang, F. Wei, S. Huang, M. Zhou, T. Zhao // Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). — 2018. — с. 654—663.

47. Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond / R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang [и др.] // arXiv e-prints. — 2016. — arXiv—1602.

48. See A., Liu P. J., Manning C. D. Get To The Point: Summarization with Pointer-Generator Networks // Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). — 2017. — с. 1073—1083.

49. Deep Communicating Agents for Abstractive Summarization / A. Celikyilmaz, A. Bosselut, X. He, Y. Choi // arXiv e-prints. — 2018. — arXiv—1803.

50. Paulus R., Xiong C., Socher R. A Deep Reinforced Model for Abstractive Summarization // International Conference on Learning Representations. — 2018.

51. Gehrmann S., Deng Y., Rush A. M. Bottom-Up Abstractive Summarization // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. — 2018. — с. 4098—4109.

52. Narayan S., Cohen S. B., Lapata M. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. — 2018. — с. 1797—1807.

53. Liu Y., Lapata M. Text Summarization with Pretrained Encoders // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). — 2019. — с. 3721—3731.

54. Unified Language Model Pre-training for Natural Language Understanding and Generation / L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, H.-W. Hon // arXiv e-prints. — 2019. — arXiv—1905.

55. Medhat W., Hassan A., Korashy H. Sentiment analysis algorithms and applications: A survey // Ain Shams engineering journal. — 2014. — т. 5, № 4. — с. 1093—1113.

56. Yaakub M. R., Latiffi M. I. A., Zaabar L. S. A Review on Sentiment Analysis Techniques and Applications // IOP Conference Series: Materials Science and Engineering. т. 551. — IOP Publishing. 2019. — с. 012070.

57. Yaakub M. R., Latiffi M. I. A., Zaabar L. S. A Review on Sentiment Analysis Techniques and Applications // IOP Conference Series:

Materials Science and Engineering. т. 551. — IOP Publishing. 2019. — с. 012070.

58. Sentiment analysis using deep learning approaches: an overview / O. Habimana, Y. Li, R. Li, X. Gu, G. Yu // Science China Information Sciences. — 2020. — т. 63, № 1. — с. 1—36.

59. The general inquirer: A computer system for content analysis and retrieval based on the sentence as a unit of information / P. J. Stone, R. F. Bales, J. Z. Namenwirth, D. M. Ogilvie // Behavioral Science. — 1962. — т. 7, № 4. — с. 484.

60. Lexicon-based methods for sentiment analysis / M. Taboada, J. Brooke, M. Tofiloski, K. Voll, M. Stede // Computational linguistics. — 2011. — т. 37, № 2. — с. 267—307.

61. Pang B., Lee L., Vaithyanathan S. Thumbs up? Sentiment Classification using Machine Learning Techniques // EMNLP. — 2002.

62. Peng W., Park D. H. Generate adjective sentiment dictionary for social media sentiment analysis using constrained nonnegative matrix factorization // Proceedings of the International AAAI Conference on Web and Social Media. т. 5. — 2011.

63. See A., Liu P. J., Manning C. D. Get To The Point: Summarization with Pointer-Generator Networks // Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). — 2017. — с. 1073—1083.

64. Content Sharing in Conflictual Ad-Hoc Twitter Discussions: National Patterns or Universal Trends? / S. Bodrunova, A. Smoliarova, I. Blekanov, A. Litvinenko // Communications in Computer and Information Science. — 2017.

65. Bag of Tricks for Efficient Text Classification / A. Joulin, E. Grave, P. Bojanowski, T. Mikolov // arXiv preprint arXiv:1607.01759. — 2016.

66. Angelov D. Top2Vec: Distributed Representations of Topics // arXiv e-prints. — 2020. — arXiv—2008.

67. Learning Cross-Lingual Sentence Representations via a Multi-task Dual-Encoder Model / M. Chidambaram, Y. Yang, D. Cer, S. Yuan, Y. Sung, B. Strope, R. Kurzweil // Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019). — 2019. — с. 250—259.

68. UMAP: Uniform Manifold Approximation and Projection / L. McInnes, J. Healy, N. Saul, L. Großberger // Journal of Open Source Software. — 2018. — т. 3, № 29. — с. 861.

69. McInnes L., Healy J., Astels S. hdbscan: Hierarchical density based clustering // Journal of Open Source Software. — 2017. — т. 2, № 11. — с. 205.