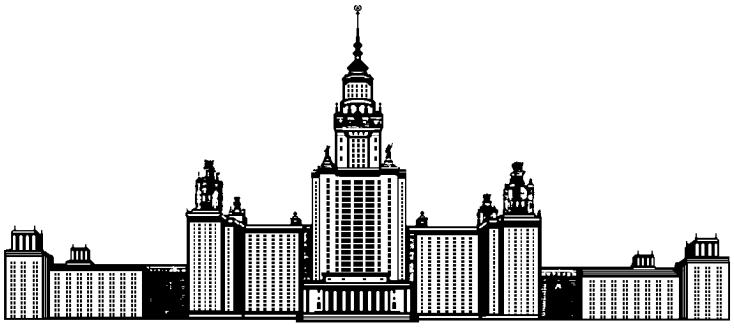
Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной Математики и Кибернетики

Кафедра Математических Методов Прогнозирования

Морозов Ярослав Олегович

Выделение трендов в коллекциях научных статей

Выпускная квалификационная работа

Научный руководитель:

д. ф.-м. н. *Воронцов К. В.*

Москва, 2023

1 Аннотация

Классические подходы к решению задачи выделения трендов в большинстве своем не используют нейросети, а также отдают большую роль графу цитирований. Однако не всегда мы имеем доступ к полной картине ссылок. Также, хочется внедрить использование глубокого обучения в решение этой задачи. В данной работе представлен метод, решающий эту задачу и опирающаяся только на текст документов. Данный метод позволяет тонко настраивать процесс выделения трендов. Также дополнительна была получена модель-трансформер решающая задачу выделения ключевых фраз. Итоговое качество выделение трендов оказывается во многом не хуже, а где-то и лучше существующих методов.

Содержание

2 Введение

В современном быстро развивающемся мире научное сообщество проводит беспрецедентный объем исследований. В результате становится все более важным эффективно выявлять и анализировать ключевые тенденции и закономерности в обширном пространстве научной литературы. Это становится особенно важно, учитывая экспоненциальный рост количества научных публикаций. Выявление наиболее актуальных и результативных тем становится титанической задачей для исследователей. Разработка эффективных методов выявления трендов в научных статьях имеет важное значение для преодоления информационной перегрузки и обеспечения возможности целенаправленного и обоснованного принятия решений. Однако перед этим стоит объяснить, что вообще такое тренд. “Радикально новая и относительно быстрорастущая исследовательская тема, характеризующаяся определенной степенью согласованности и значительным научным воздействием” (Wang, 2017).

Важность выявления тенденций в исследовательских работах можно проиллюстрировать несколькими ключевыми примерами. Например, признание появления искусственного интеллекта и машинного обучения в качестве доминирующей тенденции не только произвело революцию в технологиях, но и положило начало междисциплинарному сотрудничеству в таких областях, как биология, медицина и социальные науки. Аналогичным образом, стремительный рост количества исследований в области возобновляемых источников энергии привел к появлению инновационных подходов к решению проблем изменения климата[1] и глобальных потребностей в энергии[2]. Более того, раннее выявление тенденций может помочь выявить потенциальные сдвиги парадигмы в научном понимании, такие как открытие технологии редактирования генов CRISPR-Cas9 [3][4], которая значительно изменила область генетики и открыла новые возможности для лечения генетических расстройств.

Кроме того, алгоритмы определения тенденций предоставляют исследователям уникальную возможность открывать популярные направления исследований в других научных областях, способствуя междисциплинарным исследованиям и инновациям. Выявляя наиболее заметные тренды в различных дисциплинах, эти алгоритмы могут эффективно ликвидировать разрыв в знаниях между разрозненными областями, прокладывая путь для новаторских исследований на пересечении множества научных областей.

Данная работа направлена на удовлетворение этой важнейшей потребности путем разработки специального алгоритма для извлечения и понимания возникающих трендов в исследо-

вательских работах по различным дисциплинам. Она призвана внести вклад в оптимизацию исследовательских усилий, способствовать междисциплинарному сотрудничеству и ускорить научные открытия.

3 Постановка задачи

В рамках данной работы необходимо разработать метод выделения трендов в коллекциях научных статей.

- Алгоритм не должен уступать существующим методами
- У конечного пользователя должна быть возможность тонкой настройки, что именно считается трендом
- Алгоритм должен определять не только сам тренд, но и статью, его задавшую

4 Обзор существующих решений

Большинство существующих решений опираются на понятие “ключевая фраза”, поэтому для начала будет рассмотрены методы выделения ключевых фраз. Сам по себе термин определяет набор слов, представляющих суть всего написанного в тексте.

4.1 Выделение ключевых фраз

4.1.1 RAKE

Алгоритм быстрого автоматического извлечения ключевых слов (Rapid Automatic Keyword Extraction, RAKE)[5] — само обучаемый метод, используемый для извлечения ключевых слов из текстовых документов. Алгоритм RAKE работает путем определения и ранжирования ключевых слов-кандидатов на основе их релевантности тексту, не требуя каких-либо внешних ресурсов или уже существующих знаний. Представить его работу можно следующим образом:

1. Предварительная обработка текста: Этот шаг включает в себя очистку и нормализацию текста. Он включает в себя такие задачи, как преобразование текста в нижний регистр, удаление знаков препинания и преобразование текста в слова или фразы.

2. Удаление стоп-слов: Стоп-слова — это распространенные слова, которые не передают значимой информации (предлоги, союзы, местоимения и т. д.). На данном шаге происходит идентификация и удаление этих слов из текста, поскольку они могут негативно повлиять на процесс извлечения ключевых слов.

3. Генерация ключевых фраз-кандидатов: Генерируются ключевые фразы-кандидаты, с помощью разбиения текста на фразы в позициях стоп-слов. В результате получается список фраз, которые потенциально могут быть ключевыми словами.

4. Оценка ключевых слов: Присваивается оценка каждому ключевой фразе-кандидату на основе его релевантности тексту. Оценка рассчитывается с использованием частоты и степени употребления слов. Частота означает количество раз, когда слово встречается в тексте, в то время как степень представляет собой количество раз, когда слово встретилось в выделенных ключевых фразах. Итоговый балл вычисляется путем деления суммы степеней всех слов в ключевой фразе на сумму их частот.

5. Рейтинг ключевых фраз: Ключевые фразы-кандидаты ранжируются в порядке убывания на основе их оценок. Ключевые фразы, занявшие первое место, считаются наиболее релевантными и репрезентативными для данного текста.

Рассмотрим подробнее работу алгоритма на примере.

Compatibility of systems of linear constraints over the set of natural numbers

Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types of systems and systems of mixed types.

Manually assigned keywords:
linear constraints, set of natural numbers, linear Diophantine equations, strict inequations, nonstrict inequations, upper bounds, minimal generating sets

Рис. 1: Пример аннотации и ключевых фраз, выделенных человеком

Удаление стоп-слов, а также использование таких разделителей как запятая, точка и т.д., разбивает текст на фразы, которые считаются кандидатами в ключевые фразы.

Compatibility – systems – linear constraints – set – natural numbers – Criteria – compatibility – system – linear Diophantine equations – strict inequations – nonstrict inequations – Upper bounds – components – minimal set – solutions – algorithms – minimal generating sets – solutions – systems – criteria – corresponding algorithms – constructing – minimal supporting set – solving – systems – systems

Рис. 2: Кандидаты в ключевые фразы, полученные из аннотации

Кандидат linear Diophantine equations идет после стоп-слова of и заканчивается запятой. Следующее слово strict является началом нового кандидата в ключевые фразы strict inequations.

Далее, идет подсчет частоты и степени каждого из слов.

	algorithms	bounds	compatibility	components	constraints	constructing	corresponding	criteria	diophantine	equations	generating	inequations	linear	minimal	natural	nonstrict	numbers	set	sets	solving	strict	supporting	system	systems	upper
deg(w)	3	2	2	1	2	1	2	2	3	3	3	4	5	8	2	2	2	6	3	1	2	3	1	4	2
freq(w)	2	1	2	1	1	1	1	2	1	1	1	2	2	3	1	1	1	3	1	1	1	1	1	4	1
deg(w) / freq(w)	1.5	2	1	1	2	1	2	1	3	3	3	2	2.5	2.7	2	2	2	2	3	1	2	3	1	1	2

Рис. 3: Кандидаты в ключевые фразы, полученные из аннотации

Затем, формируется рейтинг ключевых фраз исходя из статистик слов внутри них.

minimal generating sets (8.7), linear diophantine equations (8.5), minimal supporting set (7.7), minimal set (4.7), linear constraints (4.5), natural numbers (4), strict inequations (4), nonstrict inequations (4), upper bounds (4), corresponding algorithms (3.5), set (2), algorithms (1.5), compatibility (1), systems (1), criteria (1), system (1), components (1),constructing (1), solving (1)

Рис. 4: Кандидаты в ключевые фразы, полученные из аннотации

4.1.2 Textrank

TextRank[6] - это основанный на графах алгоритм для задач обработки естественного языка, таких как извлечение ключевых слов и суммаризация текста. TextRank основан на алгоритме PageRank[7], используемом Google для ранжирования веб-страниц, и использует аналогичный подход для ранжирования текстовых элементов.

TextRank работает по следующим правилам:

1. Предварительная обработка текста: Этот шаг включает в себя очистку и нормализацию текста. Он включает в себя такие задачи, как преобразование текста в нижний регистр, удаление знаков препинания и преобразование текста в слова или фразы.

2. Построение графа: Строится граф, вершины которого представляют текстовые элементы, такие как слова и/или фразы и/или предложения. Ребра между вершинами создаются на основе их сходства.

3. Расчет сходства: Для обобщения текста сходство между двумя предложениями обычно вычисляется с использованием косинусной меры близости, которая учитывает общие слова или понятия между предложениями. Для извлечения ключевых слов ребра между двумя сущностями проводятся, если они находятся в пределах заданного окна (количество слов между ними меньше заданного числа).

4. Ранжирование графа: Алгоритм TextRank итеративно присваивает оценки вершинам (текстовым элементам) на основе их связей с другими вершинами графа. Этот процесс повторяется до тех пор, пока оценки не сойдутся и график не достигнет устойчивого состояния. Формула для пересчета выглядит так:

$$S(v_i) = (1-d) + d \sum_{v_j \in In(v_i)} \frac{w_{ji}}{\sum\limits_{v_k \in Out(v_j)} w_{jk}} S(v_j) \tag{1}$$

Где

- $S(v_i)$ — оценка (score) вершины v_i
- w_{ij} — вес ребра между вершиной i и вершиной j
- d — коэффициент затухания. Обычно выставляется равным 0.85. Определяет степень влияния соседних вершин на оценку текущей.
- $In(v_i)$ — множество вершин, из которых идут ребра в вершину v_i
- $Out(v_i)$ — множество вершин, в которые есть исходящие ребра из вершины v_i

5. Извлечение результата: При извлечении ключевых слов слова с самым высоким рейтингом на графике рассматриваются как наиболее важные ключевые слова. Для суммаризации текста извлекаются предложения с наивысшим рейтингом и объединяются, чтобы сформировать краткое изложение входного текста.

4.1.3 KeyBERT

KeyBERT[8] - современный алгоритм извлечения ключевых фраз, который использует языковые модели на основе transformer[9], в частности модель BERT (Bidirectional Encoder Representations from Transformers)[10].

Алгоритм выполняет следующие действия:

1. Получение представления текста: Текст подается в предобученную модель BERT для генерации его векторного представления. Эти представления фиксируют как семантическую, так и синтаксическую информацию текста.

2. Генерация ключевых фраз-кандидатов: KeyBERT выбирает набор ключевых фраз-кандидатов из текста. Это можно сделать, используя различные подходы, такие как n-граммы или словосочетания с существительными.

3. Получение представлений: Ключевые слова-кандидаты пропускаются через ту же модель BERT для генерации их представлений. Они обеспечивают расширенное представление ключевых фраз-кандидатов в контексте входного текста.

4. Ранжирование по ключевым фразам: Сходство между представлением текста и ключевыми фразами рассчитывается с использованием косинусного подобия или других

показателей расстояния. Ключевые фразы-кандидаты ранжируются на основе их показателей сходства, причем более высокие баллы указывают на более высокую релевантность.

5. Выбор ключевых фраз: В качестве конечного результата выбираются ключевые фразы с наивысшим рейтингом, представляющие наиболее важные и контекстуально релевантные ключевые слова для входного текста.

KeyBERT особенно полезен при работе с крупномасштабными данными и когда решающее значение имеет контекстно-зависимое извлечение ключевых слов. Он широко используется в таких приложениях, как поисковая оптимизация, рекомендации по контенту, обобщение текста и тематическое моделирование. Алгоритм также может быть точно настроен для конкретных областей или задач с помощью предварительно обученных моделей BERT для конкретной области или обучения модели на пользовательском наборе данных.

4.2 Выделение трендов

4.2.1 Citation network analysis

Анализ сети цитирований (Citation network analysis)[11] - исследование, предоставляющее метод выявления и мониторинга новых исследовательских трендов путем анализа сетей цитирования и использования методов кластеризации. Основные этапы метода можно резюмировать следующим образом:

1. Сбор данных: Собирается набор научных публикаций, включая метаданные, такие как авторы, даты публикации, названия, ключевые слова и информация о цитировании.

2. Построение сети цитирований: Создается сеть цитирований на основе собранных данных, где узлы представляют научные публикации, а направленные ребра представляют ссылки между ними.

3. Предварительная обработка: Очищаются и предварительно обрабатываются данные о цитировании, с помощью удаления самоцитирований и изолированных публикаций (публикации без каких-либо связей с цитированием).

4. Сетевой анализ: Анализируются свойства сети цитирований, такие как распределение степеней, коэффициент кластеризации, средняя длина пути и модулярность, чтобы получить представление о структуре и организации предметной области исследования.

5. Кластеризация: Применяются алгоритмы кластеризации к сети цитирований, такие как иерархическая кластеризация[12], обнаружение структуры сообществ (community structure)[13] или методы, основанные на модулярности[14], для идентификации групп тесно связанных публикаций.

6. Кластерный анализ: Анализируются полученные кластеры, чтобы определить общие темы, ключевые слова и влиятельные публикации внутри каждой группы. Исследуется хронологическое распределение публикаций внутри кластеров, чтобы оценить возникновение и развитие исследовательских тенденций.

7. Выявление трендов: Изучаются кластеры и их свойства, чтобы определить новые тренды в исследованиях. Ищутся кластеры с более высокой долей недавних публикаций, быстрым ростом размера кластера или повышением цитируемости, что может указывать на появление новых тем исследований или изменение их направленности.

8. Валидация и мониторинг: Подтверждаются выявленные тренды с использованием внешних источников данных, таких как мнения экспертов, и идет слежка за их развитием с течением времени, с периодическим обновлением сети цитирований, а также с повторением этапов кластеризации и идентификации трендов.

4.2.2 Incremental Topic Model

Инкрементное тематическое моделирование (Incremental Topic Model)[15] — это подход, используемый для эффективной адаптации тематических моделей к новым поступающим данным. Это особенно полезно при работе с крупномасштабными и постоянно растущими наборами данных, такими как базы данных научной литературы. Традиционные методы тематического моделирования, такие как латентное размещение Дирихле (Latent Dirichlet Allocation, LDA)[16], часто требуют переподготовки с нуля при добавлении новых данных, что может быть дорогостоящим с точки зрения вычислений и отнимает много времени. Основная идея инкрементного тематического моделирования заключается в обновлении тематической

модели по мере поступления новых данных, без необходимости переподготовки модели с нуля. Существует несколько подходов для достижения этой цели, таких как online LDA[17], являющийся расширением стандартного LDA, которое может обновлять модель в режиме онлайн по мере добавления новых документов в набор данных.

1. Инициализация: Дана коллекция документов D и словарь слов W. Инициализируется тематическая модель исходным набором тем T. Распределения слов по темам представляется в виде матрицы Φ, распределение документов по темам с помощью матрицы Θ.

2. Новая коллекция документов: При появлении новой коллекции документов D', анализируется набор появляющихся слов W' и обновляются текущие темы T, добавлением новых тем T'.

3. Инкрементное обновление: Изменяются матрицы Φ и Θ добавлением новых строк и столбцов, связанных с временными метками и новой коллекцией документов.

4. Определение количество новых тем: Рассчитывается словарь новых трендов V, состоящий из терминов, которые стали гораздо чаще использоваться с момента последнего обновления. Определяется количество новых тем $|T'|$. Для текущей временной метке слово добавляется к словарю V, если оно встречается по крайней мере в документах $mindf$ и удовлетворяет условию тренда:
$$\frac{tf_{new} - tf_{load}}{tf_{old}} > \alpha \tag{2}$$
, где tf_{old} - количество встречаемости w в документах D, а tf_{new} - количество встречаемости w в $D \cup D'$. α - гиперпараметр регулирования, который задает степень увеличения встречаемости слов для классификации их как трендовых. Количество тем определяется как
$$|T'| = |T_{start}| + \lfloor \frac{|V|}{\beta} \rfloor \tag{3}$$
, где T_{start} определяет количество тем в начальной временной метке, $\beta \in \mathbb{N}$ ограничивает количество добавляемых тем.

5. Определение тренда: Для каждого возникающего тренда находится наиболее подходящая тема на основе показателя отзыва Recall@k как для документов, так и для слов.
$$XRecall@k = \frac{|X_{topic}[:k] \cap X_{trend}|}{k} \tag{4}$$

5 Выделение именованных сущностей

Выделение именованных сущностей (Named Entity Recognition, NER)[18] является важнейшей задачей в обработке естественного языка (Natural Language Processin, NLP), целью которой является идентификация и категоризация объектов реального мира, таких как лица, организации, местоположения, даты и количества, встроенные в неструктурированные текстовые данные. С быстрым ростом объема текстовой информации, доступной в цифровом виде, NER стал важным компонентом различных приложений NLP, включая извлечение информации, обобщение текста, анализ настроений и вопросно-ответных систем.

Задача NER может быть определена как задача маркировки последовательности, где каждому слову или токену в тексте присваивается метка, соответствующая определенному классу сущностей или классу, соответствующему отсутствию какой-либо сущности у данного объекта. Модель распознавания должна уметь точно предсказывать метки для входного текста. Модели NER должны решать различные задачи, такие как устранение неоднозначности сущностей, многословные сущности, вложенные сущности и объекты, зависящие от домена.

С задачей NER связан ряд проблем, в том числе:

- Неоднозначность: Сущности могут иметь одно и то же имя, что приводит к неоднозначности в определении правильного класса сущностей. Например, “Apple” может относиться к компании или фрукту.
- Многословные сущности: Сущности могут охватывать несколько слов, и модели NER должны научиться идентифицировать и классифицировать эти фразы как единую сущность.
- Вложенные объекты: Объекты могут быть вложены в другие объекты, что усложняет задачу маркировки.
- Объекты, относящиеся к конкретной предметной области: Объекты, относящиеся к конкретной предметной области, требуют специальных знаний и могут быть неточно распознаны NER-моделями общего назначения.

Методы и приемы, используемые в задаче NER, значительно эволюционировали с течением времени, следующим образом:

1. Методы, основанные на правилах (Rule-Based methods)[19]: Ранние системы NER полагались на разработанные вручную правила и шаблоны для идентификации и классификации объектов. Эти системы были ограничены их неспособностью обобщаться на новые объекты и домены.

2. Машинное обучение на основе характеристик (Feature-based methods)[20]: С появлением методов машинного обучения к более сложным задачам стали применяться модели, основанные на характеристиках объектов, такие как деревья решений, метод опорных векторов и модели максимальной энтропии. Эти модели требовали ручной разработки функций для сбора релевантной лингвистической и контекстуальной информации.

3. Модели последовательностей (Sequence models): Модели последовательностей, такие как скрытые марковские модели (Hidden Markov Model, HMM)[21] и условные случайные поля (Conditional Random Fields, CRF)[22], были введены для определения зависимостей между соседними токенами в текстовой последовательности, повышая производительность систем NER.

4. Глубокое обучение: С появлением глубокого обучения архитектуры нейронных сетей, такие как рекуррентные нейронные сети (Recurent Neural Networks, RNN), сети с длительной кратковременной памятью (Long Short-Term Memory, LSTM)[23] и двунаправленные LSTM[24], стали использоваться для более сложных задач. Эти модели могут автоматически изучать представления объектов и фиксировать сложные зависимости в тексте.

5. Трансформеры: Совсем недавно методы переноса обучения и предварительно обученные языковые модели, такие как BERT (Bidirectional Encoder Representations from Transformers), GPT(Generative Pre-trained Transformer)[25] и RoBERTa(A Robustly Optimized BERT Pretraining Approach)[26], значительно улучшили производительность систем NER. Эти модели могут быть точно настроены для конкретных задач NER, используя богатые контекстуальные представления, полученные на основе крупномасштабных объемов данных.

Математически, задача NER может быть определена как задача поиска специальной функции, переводящей последовательность токенов в последовательность выходных меток.

$$f(x_1, x_2, \ldots, x_n) = (y_1, y_2, \ldots, y_n)$$

, где $x_i \in X$ — токен из заранее заданного словаря, $y_i \in Y$ - метка токена. Чаще всего используют BIO-схему разметки. B - begin (начало именованной сущности), I - inside (состояние внутри именованной сущности), O-outside (данный токен не принадлежит ни одному из классов сущностей). В свою очередь, метки B и I могут распадаться на множество подметок, то есть тегов, в зависимости от типа задачи. В общем случае, эти теги могут быть:

- PER (Person), соответствующие метки B-PER, I-PER
- LOC (Location), соответствующие метки B-LOC, I-LOC
- ORG (Organization), соответствующие метки B-ORG, I-ORG

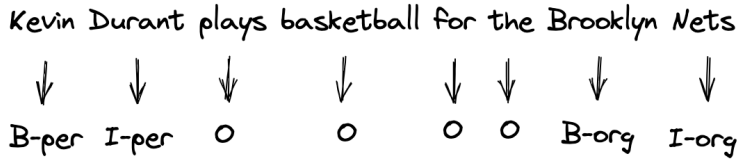


Рис. 5: Пример кодирования предложения в задаче NER

Конкретные теги зависят от специфики задачи и желаний исследователя.

Обучение нейросетей для задачи NER в целом походит на обучение любой модели для классификации. Теперь будем считать, что наша функция $f(x_1, x_2, \ldots, x_n)$ выдает не сразу метки классов, а вероятностное распределение на них, то есть

$$f(x_1, x_2, \ldots, x_n) = (z_1, z_2, \ldots, z_n),\ z_i \in \mathbb{R}^{2N+1},\ \sum_{j=1}^{2N+1} z_i^j = 1,\ z_i^j \geq 0\ \forall j \in \{1,\ \ldots,\ 2N+1\}$$

, где z_i^j - j-ый элемент i-го выходного вектора вероятностей классов токена x_i. Размерность вектора обуславливается количеством тегов (N) - по два (B и I) на каждый, а также метка O. Соответственно получать итоговую метку токена можно как простым взятием аргмаксимума, так и множеством других методов, доступных при наличии вероятностного распределения. При обучении на вход модели подается последовательность токенов и их разметка. Обучение проходит стандартным способом - методом обратного распространения ошибки.

Обратное распространение[27], сокращенно от “обратного распространения ошибок”, является широко используемым алгоритмом оптимизации для обучения искусственных нейронных сетей. Это контролируемый метод обучения, который регулирует веса соединений в сети, чтобы свести к минимуму разницу между прогнозируемым выходом и фактическим выходом (также известной как ошибка или потеря). Алгоритм обратного распространения состоит из двух основных этапов:

- Прямой переход: На этом этапе входные данные передаются по сети для вычисления прогнозируемого выходного сигнала. Входные данные распространяются по сети слой за слоем, применяя веса и функции активации, пока не будут сгенерированы выходные данные.

- Обратный проход: На этом шаге ошибка между прогнозируемым выходом и фактическим выходом вычисляется с использованием функции потерь. Градиент этой ошибки по отношению к каждому весу вычисляется с использованием цепочного правила для производных, которое является ключом к алгоритму обратного распространения. Затем градиенты используются для обновления весов в сети, чтобы свести к минимуму ошибку. Процесс выполняется в обратном порядке, начиная с выходного слоя и продвигаясь к входному слою.

Итеративно выполняя прямые и обратные проходы, нейронная сеть учится корректировать свои веса, чтобы минимизировать ошибку, что позволяет ей делать более точные прогнозы.

В качестве функции ошибок для задачи NER выступает потеря перекрестной энтропии (Cross-Entropy Loss, CEL)[28]. CEL определяет разницу между предсказанным вектором вероятности моделью и истинными метками. Математически кросс-энтропия определяется как:

$$L(p, y) = -\sum_{i} y \cdot \log p_i \cdot \mathbb{I}[i = y] \tag{5}$$

, где y - истинная метка класса, p - предсказанный вектор вероятностей.

6 Модель выделения трендов

Модель выделения сущностей представляет собой соединение двух этапов: выделение ключевых фраз документа, которые теоретически уже являются трендами, затем постобработка полученных данных.

6.1 Выделение слов-трендов

6.1.1 Перенос обучения

Переносное обучение (Transfer Learning)[29] — это метод машинного обучения, при котором предварительно обученная модель, которая уже изучила полезные функции или представления из большого набора данных, адаптируется для решения связанной, но отдельной задачи. Идея переноса обучения заключается в том, чтобы использовать знания, полученные в ходе первоначального обучения, для повышения эффективности выполнения новой задачи, часто при ограниченных данных или ресурсах. Переносное обучение оказалось особенно эффективным для моделей на основе трансформеров в задачах обработки естественного языка. Трансформеры, такие, как BERT, GPT и RoBERTa, предварительно обучаются работе с массивными наборами данных и изучают широкий спектр лингвистических функций, которые могут быть точно настроены для конкретных задач с меньшими помеченными наборами данных. Такой подход имеет несколько преимуществ:

- Улучшенная производительность: Перенос обучения позволяет модели опираться на богатые представления, полученные во время предварительной подготовки, что часто приводит к повышению производительности при выполнении целевой задачи по сравнению с обучением модели с нуля.

- Более быстрое обучение: Поскольку начальные слои модели уже изучили значимые функции, точная настройка предварительно обученной модели часто требует меньшего количества периодов обучения, что приводит к более быстрому времени обучения.

- Обобщение: Предварительно обученные модели-трансформеры отражают общие лингвистические особенности, которые полезны в широком спектре задач, что делает их легко адаптируемыми к различным областям и проблемам.

Особенно хорошо используется перенос обучения для специфичных доменов задач, например для работы с научными документами. Модели, обученные на общих текстах имеют хорошие знания и контекстуальные представления слов в общем виде. Однако при использовании таких моделей напрямую в специфичной задаче результат может оказаться плачевным. Дообучение такой модели даже на сравнительно небольшом корпусе специфичных для задачи данных способно кратно увеличить качество ее работы.

6.1.2 Оценка качества

Чтобы понять насколько хорошей получилась модель необходимо как-то оценить ее качество. Стандартный способ в случае, когда есть набор предсказанных меток и реальных, это использование точности (precision), полноты (recall) и F-меры (f-measure). Каждая из этих метрики отражает какой-то свой смысл получившегося результата. Для начала введем следующие понятия:

- True Positives (TP) - количество верно определенных ключевых фраз, которые модель предсказала и которые есть в наборе верных ключевых фраз.
- False Positives (FP) - количество неверно определенных ключевых фраз, которые модель предсказала, но которых нет в наборе верных ключевых фраз.
- False Negatives (FN) - количество верных ключевых фраз, которые модель не предсказала.

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$F_1 = 2\frac{Precision \cdot Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \tag{8}$$

Точность отражает способность модели правильно классифицировать только релевантные объекты среди всех объектов, которые модель классифицировала как релевантные. Полнота отражает способность модели идентифицировать все релевантные объекты среди всех релевантных объектов и ложно отрицательных объектов. F-мера является гармоническим средним между точностью и полнотой и предназначена для учета обеих метрик одновременно. Это полезно, когда нужно сбалансировать точность и полноту.

6.1.3 SPECTER2

Для выделения ключевых фраз использовалась базовая модель SPECTER2[30]. С помощью переноса обучения модель была дообучена на задачу выделения ключевых фраз в наборе научных текстов, каждая из которых содержит главный смысл всего своего документа. Дообучение проводилось на коллекциях документов с научных конференций SemEval[31].

В этом наборе данных содержатся аннотации статей, а также список ключевых фраз к ним. Одной из проблем был переход от пары (текст, ключевые фразы) к парам (последовательность токенов, последовательность NER меток). В качестве решения предлагается следующий алгоритм:

1. Токенизация текста на предложения. Например, с помощью NLTK[32]
2. Токенизация предложений на слова
3. Приведение к нижнему регистру и лемматизация слов как самих текстов, так и ключевых фраз.
4. Для каждой ключевой фразы поиск ее вхождения в предложения.
5. Создание последовательности NER меток на основе этих вхождений
6. Перенос меток на изначальный вид предложений

Algorithm 1: Переход от пар (текст - ключевые фразы) к BIO разметке

Input: Набор текстов $\mathcal{D}$ и ключевых фраз к ним $\mathcal{KW}$

Output: BIO-разметка обучающего корпуса, список токенов $Tokens$ и меток $Tags$

$Tokens \leftarrow []$

$Tags \leftarrow []$

for $d \in \mathcal{D}$ **do**

Токенизация на предложения: $S \leftarrow \texttt{SentenceTokenizer}(d)$

Токенизация на слова: $W \leftarrow \texttt{WordTokenizer}(S)$

Лемматизация слов и ключевых фраз: $W \leftarrow \texttt{Lemmatization}(W)$,

$\mathcal{KW} \leftarrow \texttt{Lemmatization}(\mathcal{KW})$

Обновление списка токенов: $Tokens \leftarrow Tokens + W$

Обновление списка меток: $Tags \leftarrow Tags + [\underbrace{O, O, \ldots, O}_{|W|}]$

for $kw \in \mathcal{KW}$ **do**

Получение позиций ключевых фраз в предложении: $I \leftarrow \texttt{KeywordFinder}(d)$

Проставление по нужным индексам правильных тегов: $\texttt{PutTags}(Tags, I, kw)$

end

end

Трансформация токенов в изначальный вид: $Tokens \leftarrow \texttt{TransformTokens}(Tokens)$

Более формально говоря, мы приводим все слова и тексты к одному нормальному виду, после чего ищем где именно в тексте находятся ключевые фразы, которые были выписаны авторами статей. Так как сами фразы и их вхождения в текст могут не совпадать по числу, форме и другим признакам слов, необходимо привезти их всех к единой форме, за что отвечает пункт 3. Затем используется простой линейный поиск ключевых фраз. Тут играет роль то, что обычно ключевые фразы состоят не более, чем из пары слов. Поэтому можно использовать наивный алгоритм. Пусть теперь мы нашли индексы вхождения ключевых фраз в предложения (имеется в виду по словам). Тогда мы имеем последовательности вида $(t_0^0, t_1^0, \ldots, t_{n_0}^0)$, $(t_0^1, t_1^1, \ldots, t_{n_1}^1)$, $\ldots$, $(t_0^k, t_1^k, \ldots, t_{n_k}^k)$, где k - количество предложений, n_i - количество токенов в i-ом предложении, $t_i^j \in \{B, I, O\}$ - метка i-го токена в j-ом предложении. Простой конкатенацией этих последовательностей получаем разметку всего текста. Отдельно стоит вопрос, нельзя ли сразу применить токенизацию на слова ко всему тексту, но практика показывает, что так результат будет хуже.

Model		Accuracy		Precision		Recall		F-measure	
Lr	Epochs	Train	Test	Train	Test	Train	Test	Train	Test
10^{-5}	12	99.6	86.7	99.6	**88.3**	99.6	86.7	99.6	**86.9**
10^{-3}	12	81.8	81.3	67.3	67.2	81.9	81.4	73.8	73.4
10^{-4}	12	**99.8**	86.1	**99.8**	87.3	**99.8**	86.1	**99.8**	86
10^{-4}	15	**99.8**	**87.1**	**99.8**	88.1	**99.8**	**86.8**	**99.8**	86.8

Таблица 1: Результаты дообучения модели SPECTER2

6.2 Сравнение методов выделения ключевых фраз

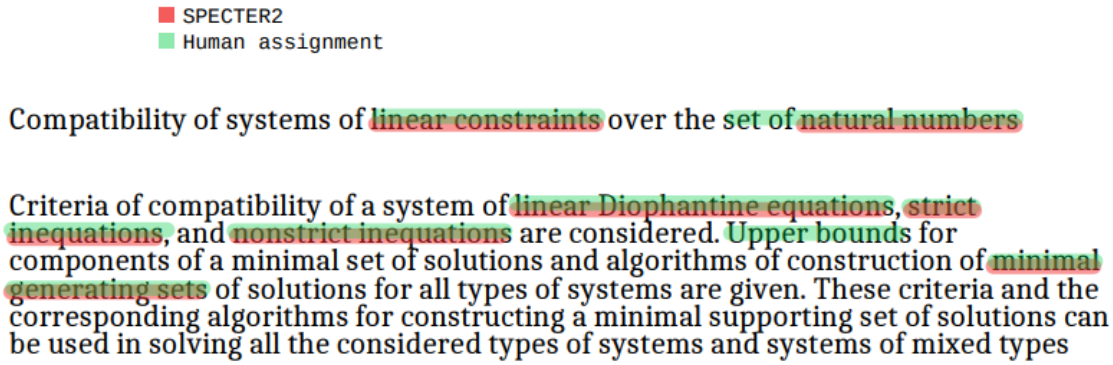


Рис. 6: Выделение ключевых фраз

Для сравнения моделей использовался предварительно обработанный набор данных для извлечения ключевых фраз Inspec[33]. Он содержит в себе аннотации к статьям с различных конференций, разделенных на обучающие данные (1000 текстов), валидационные и тестовые (по 500 текстов). К каждой из аннотаций прилагается набор ее ключевых фраз.

Так как валидация состояла не из одного текста, а из нескольких, необходимо было как то усреднить результаты метрик. Использовалась следующая стратегия:

- Посчитать для каждого текста набор метрик
- Сложить эти наборы по всем текстам
- Усреднить получившийся результат

RAKE	**Textrank**	**KeyBERT**	**SPECTER2**	**Human**
linear constraints	linear constraints	linear constraints, systems of linear constraints	linear constraints	linear constraints
linear diophantine equations	linear diophantine equations	linear diophantine equations, linear diophante	linear diophantine equations	linear diophantine equations
natural numbers	natural numbers		natural numbers	set of natural numbers
strict inequations	strict inequations		strict inequations	strict inequations
nonstrict inequations	nonstrict inequations	nonstrict inequations	nonstrict inequations	nonstrict inequations
upper bounds	upper bounds			upper bounds
minimal generating sets, minimal supporting set, minimal set			minimal generating sets	minimal generating sets

Таблица 2: Результат работы различных методов извлечения ключевых фраз

Method	Extracted keywords		Correct keywords		Precision	Recall	F-measure
	Total	Mean	Total	Mean			
RAKE($T = 0.33$)							
KA stoplist ($df > 10$)	6052	12.1	2037	4.1	33.7	41.5	37.2
TextRank							
Undirected, co-occ. window $= 2$	6784	13.6	2116	4.2	31.2	43.1	36.2
(Hulth 2003)							
Ngram with tag	7815	15.6	1973	3.9	25.2	51.7	33.9
SPECTER2							
$lr = 10^{-5}$	5410	10.8	2741	5.5	61.8	56	57.4 (58.8)
$lr = 10^{-4}$	5050	10.1	2508	5	60.4	51.5	54.1 (55.6)
BERT KW Extractor							
	4500	9	2300	4.6	52.4	42.3	45.4 (46.8)

Таблица 3: Результаты автоматического выделения ключевых фраз

При такой стратегии, однако, теряется связь между F-мерой и полнотой с точностью. Поэтому в скобках в таблице 3 для последних трех моделей указано значение F-меры, посчитанное для усредненных полноты и точности по формуле (8).

Как видно модель SPECTER2 значительно превзошла свои не нейросетевые аналоги, а также лучше подобных моделей-трансформеров, обученных на доменах общего назначения.

6.3 Определение трендов

После того как часть модели, отвечающая за выделение ключевых фраз, готова к использованию, можно переходить к непосредственному распознаванию трендов. Пусть дан корпус текстов, в которых мы хотим определить текущие тренды. Идея алгоритма состоит в следующем:

1. Пропустить каждый из текстов через модель выделения ключевых фраз
2. Собрать статистики по каждой из ключевых фраз
3. Отфильтровать множество ключевых фраз
4. Исходя из конкретных желаний объявить какие-то из фраз трендами
5. Найти первоисточник тренда

6.3.1 Статистики и фильтрация ключевых фраз

После этапа выделения ключевых фраз во всем корпусе документов нужно как-то перейти к непосредственно трендам. В данной работе предлагается это сделать при помощи следующих независимых способов:

- Подсчет количества упоминаний каждой ключевой фразы в целом во всем наборе документов. Базовая идея – чем больше какая-то ключевая фраза встречалась во всем наборе текстов, тем вероятнее, что она является трендом в этих данных. Тут же появляется два вопроса. Первый — начиная с какого количества вхождений считать, что фраза тренд? Второй, как быть с выделенными ключевыми фразами общего назначения, такими как, например, “обучающие данные” и “нейросети”. Здесь необходимо предложить механизм тонкой настройки этих параметров. Должна быть возможность передавать в модель список фраз, которые по мнению пользователя являются ключевыми фразами общей лексики. Также должна быть возможность ручной задачи параметра, отвечающего за необходимое количество вхождений. Предлагается его задавать либо в абсолютном значении, либо в относительном (чтобы фраза считалась трендом, она должна войти в хотя бы $d \in [0, 1]$ документов от общего их числа).

- Подсчет упоминаний в каждом документе по отдельности. Здесь предлагается настраивать сколько раз должна встретиться фраза в каждом из документов, чтобы она считалась трендом. Очевидно, вряд ли будет фраза не общей лексики, встречающаяся абсолютно везде, даже если она тренд. Поэтому на самом деле этот пункт идет в комбинации с предыдущим. Например, можно сказать, что фраза — тренд, если она в какой-то части документов встретилась какое-то количество раз. Также, естественно, можно задать формулу весов для документов. Возможно мы не хотим учитывать документ, в котором фраза повторяется 100 раз с документом, в котором она повторяется 5 раз, с одинаковой важностью. Тут можно либо обрезать вклад каждого документа по какому-то порогу, либо считать статистикой фразы не ее количество повторений, а отношение ее встречаемости в документе к размеру документа, а дальше уже работать с этими величинами.

- Группировка схожих ключевых фраз в одно общее понятие, то есть объединение синонимов, различных по написанию, но совпадающих по смыслу фраз (cifar10, cifar - 10) в одно целое. Тут, опять, либо можно просто сложить все статистики объектов компоненты и работать с получившимся результатом как с новым отдельным объектом, либо как-то агрегировать.

В данной работе рассмотрены не все возможные методы, что открывает возможности для дальнейших исследований.

6.3.2 Первоисточники тренда

После этапа сбора статистики и фильтрации ключевых фраз имеется набор определенных трендов, которые можно выдать конечному пользователю. Но довольно часто требуется не только распознать сам тренд, но и найти его первоисточник. Однако тут возникает проблема с непосредственно определением тренда.

На данном графике 7 можно выделить три возможных начала тренда.

- Зеленая вертикальная линия: первое упоминание ключевой фразы. Можно считать, что тренд задается статьей, в которой впервые была упомянута соответствующая ему ключевая фраза. Наклонная зеленая линия задает примерный тренд с началом в первой статье.

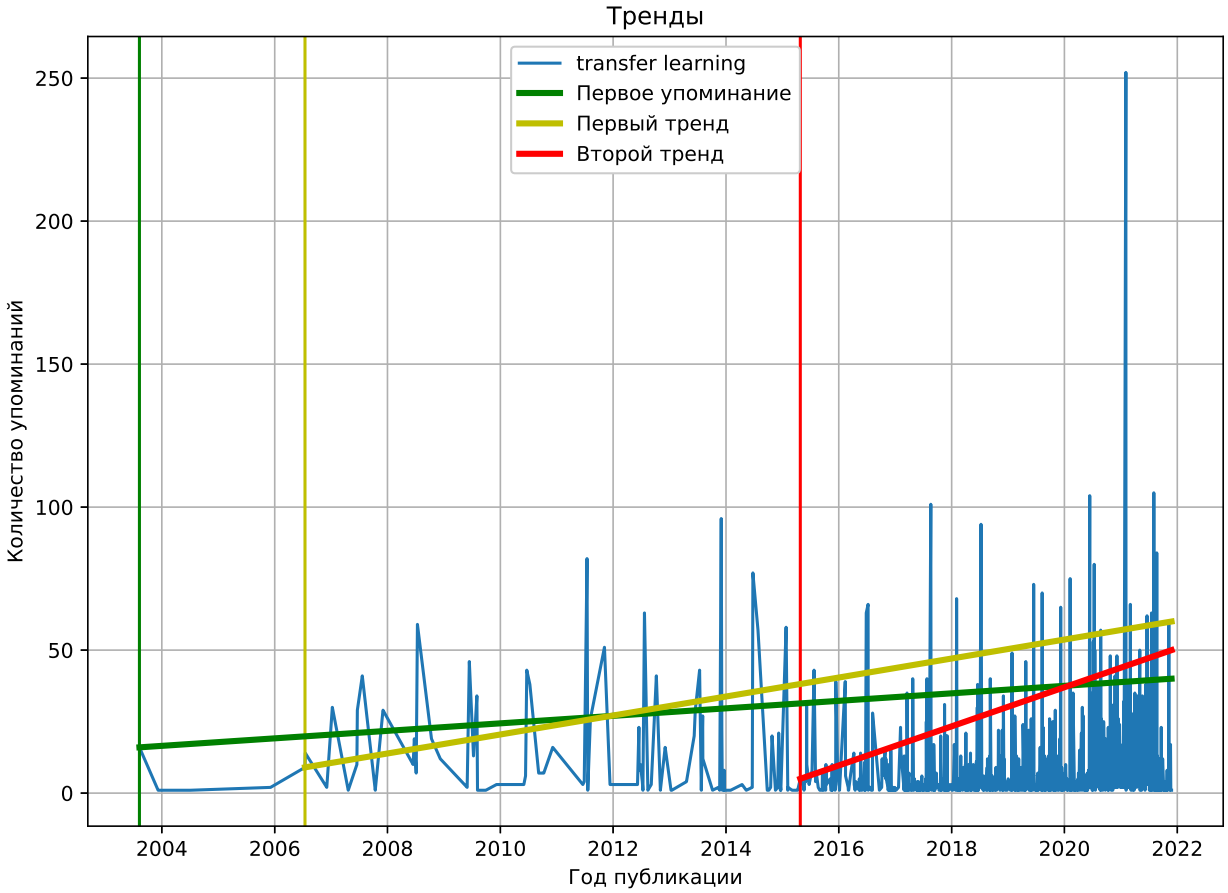


Рис. 7: Зависимость количества упоминаний ключевой фразы Transfer learning от года

- Желтая вертикальная линия: первые разы, когда ключевая фраза стала упоминаться большое количество раз. На графике желтая наклонная линия задает тренд отвечающий “внешним” точкам большого количества публикаций. Отличительные черты — резкие изменения по амплитуде, достаточно большая ширина между соседними подъемами.

- Красная вертикальная линия: начало постоянно идущих публикаций по теме. Отличительная черта — практически отсутствующие проблески, явно прослеживаемая зависимость от времени.

Даже казалось бы из такого простого графика уже очевидно какой именно тренд, (а, более конкретно — какое начало) это то, что нам нужно. В данной работе будет рассмотрен первый вариант выделения, то есть первое упоминание. Также стоит заметить, что тренды могут в какой-то момент выйти на плато (например, как сейчас, трансформеры). О них по-прежнему много пишут, но уже нельзя сказать что это что-то новое. Одним из возможных критериев

начала тренда является дата, после которой идет экспоненциальный рост количества публикаций по теме.

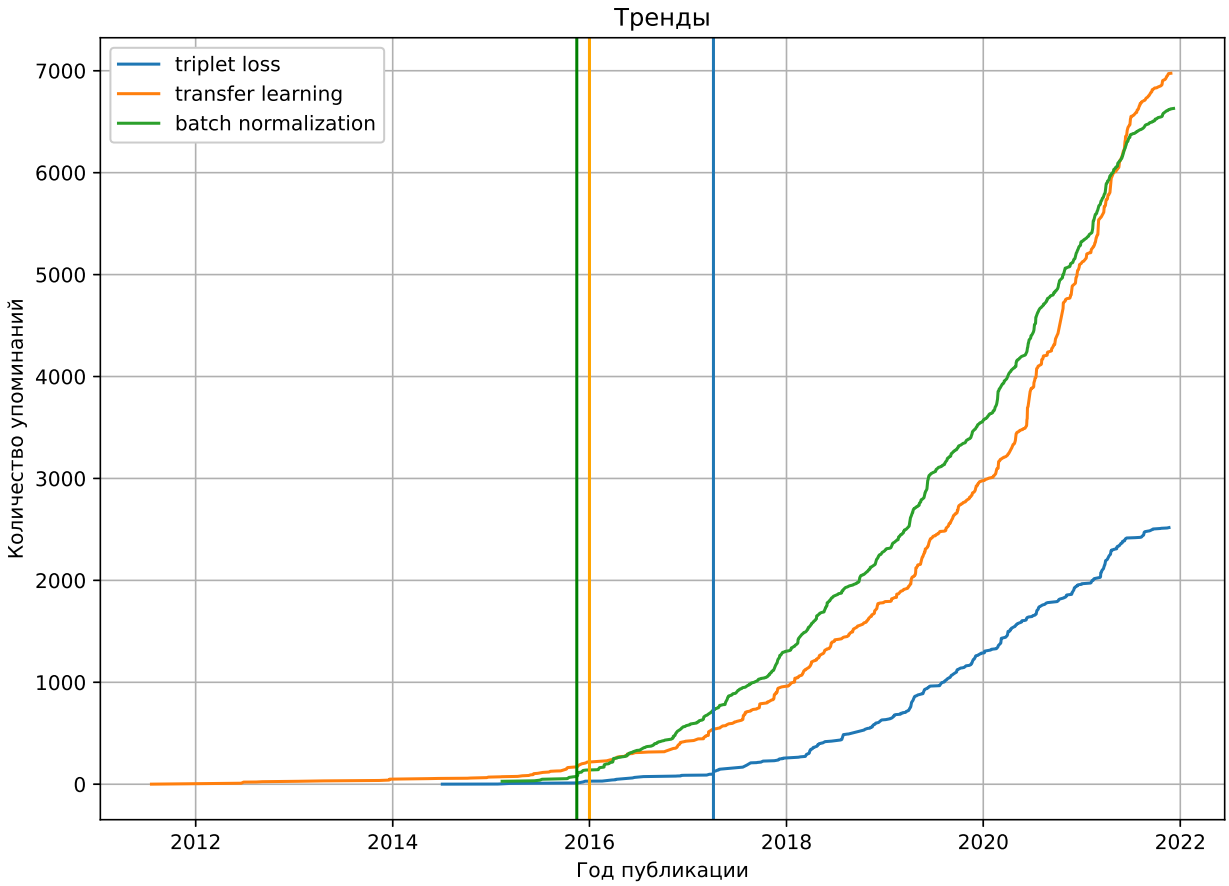


Рис. 8: Зависимость общей суммы количества упоминаний трендов от года

Здесь вертикальный линии как раз отражают начало экспоненциального роста количества упоминаний, то есть возможную дату, с которой стоит считать тренд начавшимся.

6.4 TrendSPECTERModel

Исходя из всего вышесказанного, итоговый алгоритм можно описать следующим образом:

Algorithm 2: Использование модели выделения трендов TrendSPECTERModel

Input: Обучающий корпус $\mathcal{D}$ документов

Output: Набор ключевых фраз трендов T, набор документов TD их содержащих

Создание словаря $StDct$

for $d \in \mathcal{D}$ **do**

  Извлечение ключевых фраз: $K \leftarrow$ `KeywordExtractor`(d)

  Получение статистик по ключевым фразам: $KS \leftarrow$ `GetStatistics`(K, d, D)

  Обновление словаря в соответствии с выбранной политикой обработки статистик: `UpdateStatsDict`$(StDct, K, KS)$

end

$StDct \leftarrow$ `FilterKeywordsDict`$(StDct)$

$T, TD \leftarrow$ `FindTrends`$(StDct, D)$

Модель поддерживает используемые функции в себе по умолчанию, однако можно также задавать свои, если хочется как-то обрабатывать результаты по своему. Опишем, что конкретно делали функции в реализации данной работы:

- KeywordExtractor(d). Дообученная модель SPECTER2. Выдает список вида $(k_i,\ conf_i)$. По умолчанию по итогу берет только те ключевые фразы, уверенность модели в которых превышает 0.99.

- GetStatistics(K, d, D). Считает сколько раз ключевая фраза k встретилась в наборе K. Возвращает список пар вида $(k,\ n_k)$.

- UpdateStatsDict($StDct, K, KS$). Увеличивает значения соответствующих счетчиков k в словаре $StDct$ на величину n_k.

- FilterKeywordsDict($StDct$). Убирает из словаря ключевые фразы обшей лексики.

- FindTrends($StDct, D$). Возвращает список все документов, в котором были ключевые фразы из словаря, и сами эти ключевые фразы.

Преимущество данной модели состоит в том, что она представляет собой композитную структуру, что позволяет тонко настраивать работу с выделением трендов под свои нужды.

7 Эксперименты

Для проведения экспериментов использовался набор данных, содержащий статьи из Semantic Scholar ORD. Помимо самих текстов и дат их публикаций, в нем также содержится набор трендов, определенный графом ссылок.

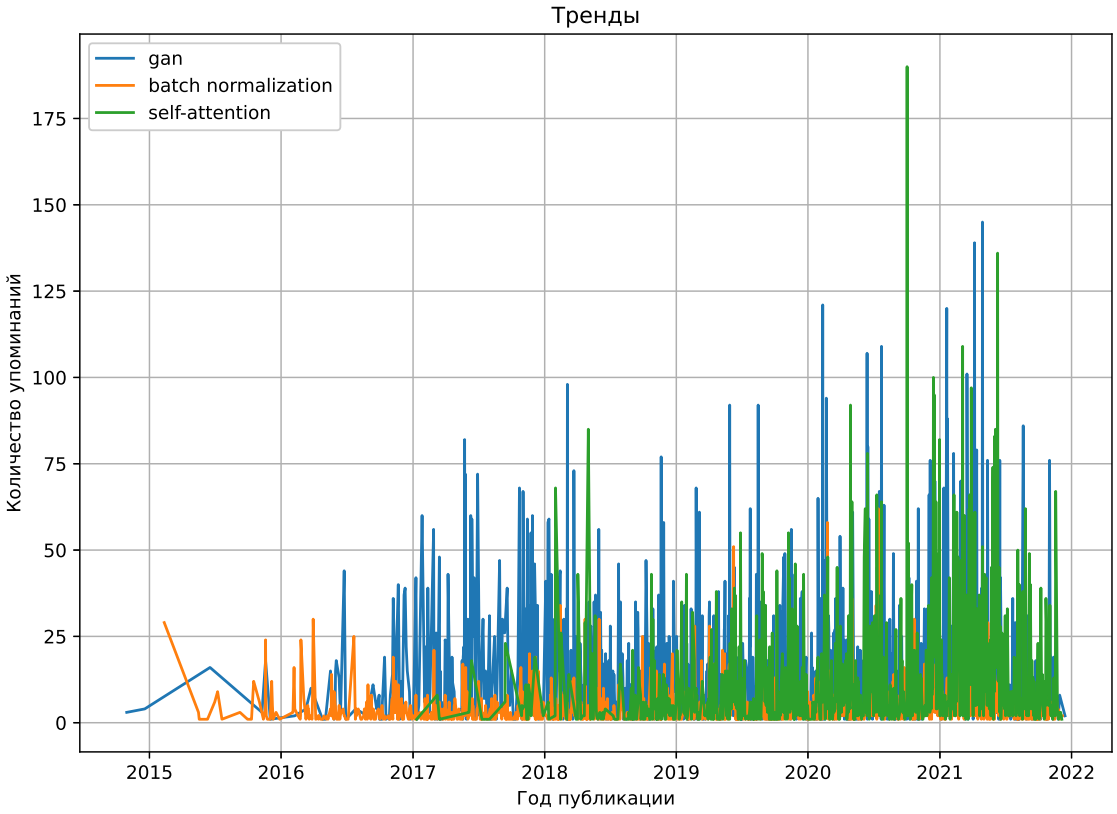


Рис. 9: Пример зависимостей количества публикаций от года для нескольких трендов, выделенных моделью TrendSPECTER

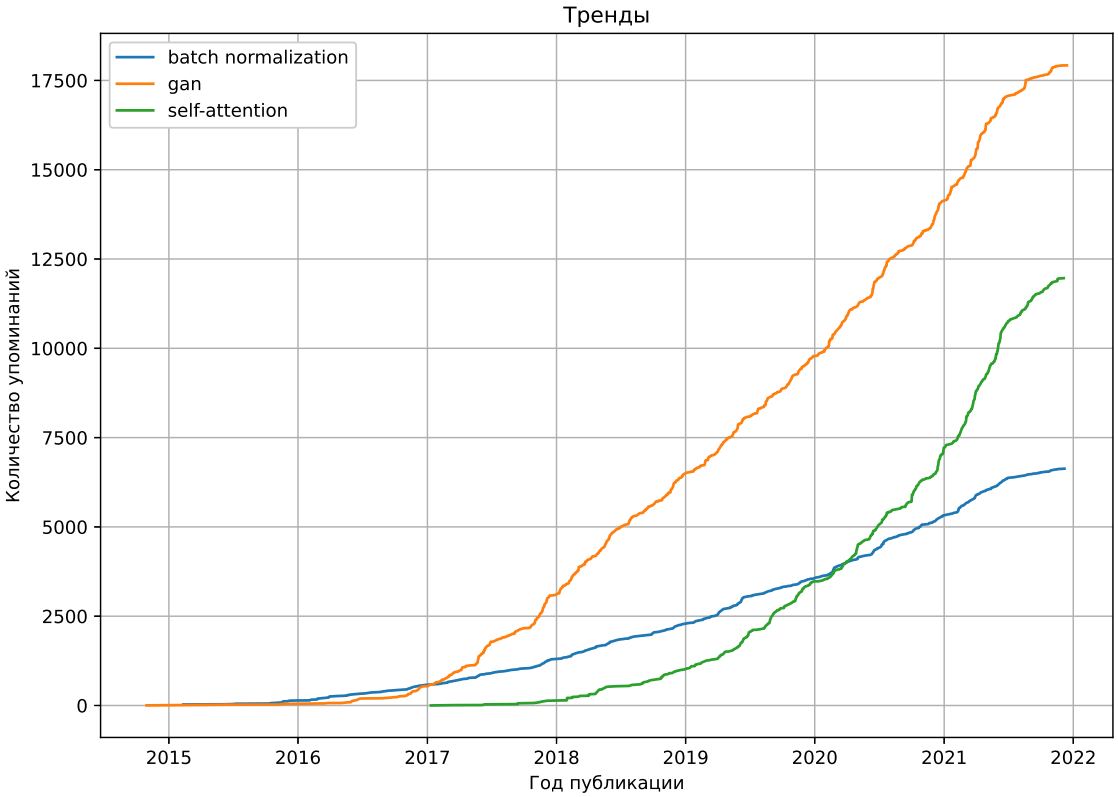


Рис. 10: Пример зависимостей количества публикаций от года для нескольких трендов, выделенных моделью TrendSPECTER

В качестве экспериментов было исследовано выделение трендов в виде самого первого упоминания и в виде точки начала экспоненциального роста. Для определения этой точки использовалась следующая эвристика: пусть $window$ - ширина окна в массиве $values$ кумулятивно просуммированных количеств упоминаний тренда по годам. Пусть гиперпараметр d - фактор, отвечающий за контролируемость начала экспоненциального роста. Тогда началом считается первый индекс j, для которого величина $\frac{values[window:]}{values[:-window]} > d$. Но это индекс в укороченном массиве. Значит реальный индекс это $window + j$. Идея очень простая. Ширина окна отвечает за то, насколько вперед мы смотрим при сравнении значений на графике. Параметр d отвечает на вопрос о том, как сильно должно вырасти количество публикаций, чтобы это можно было считать началом экспоненциального роста. Также для обеих моделей есть параметр c, отвечающий за то, сколько раз должна встретиться ключевая фраза во всем корпусе документов, чтоб она считалась трендом.

Выставляя параметр c в ноль, мы как будто говорим модели извлекать тренды независимо от их общего числа упоминаний, а лишь исходя из информации об экспоненциальных их

Params	mean	min	25%	50%	75%	max	#extracted
$d = 10, window = 5, c = 100$	146.8	**0**	**0**	54	164	1345	86
$d = 5, window = 3, c = 100$	**81.4**	**0**	**0**	47	**63**	1185	87
$d = 3, window = 3, c = 0$	95.2	**0**	**0**	**23**	72	1185	**90**
$d = 10, window = 3, c = 0$	140	**0**	**0**	45	130	**1078**	88
$d = 50, window = 10, c = 250$	467	**0**	11	206	766	2740	62
$c = 1000$	0	0	0	0	0	0	70

Таблица 4: Статистика задержки в выделении трендов в днях. Максимум, минимум и персентили по сдетектированным трендам с заданными конфигурациями.

подъемах. В качестве меры сравнения использовались различные статистики по разнице в днях между выделенными трендами и валидационным набором трендов из графа ссылок. В качестве начала тренда из валидационного датасета бралась минимальная дата публикации статьи, содержащей соответствующую ключевую фразу.

LSTM. A more recent contextualized model is **BERT** (Devlin et al., 2019). The technique is built upon earlier contextual representations, including ELMo, but differs in the fact that, unlike those models which are mainly unidirectional, BERT is bidirectional, i.e., it considers contexts on both sides of the target word during representation. We experimented with two pre-trained BERT models: *base* (768 dimensions, 12 layer, 110M parameters) and *large* (1024 dimensions, 24 layer, 340M parameters).[9] Around 22% of the pairs in the test set had at least one of their target words not covered by these models. For such out-of-vocabulary cases, we used BERT's default tokenizer to split the unknown word to subwords and computed its embedding as the centroid of the corresponding subwords' embeddings.

Рис. 11: Упоминание BERT в отредактированной статье

Интересный факт, выявленный в ходе исследований, это редактирование статей авторами с добавлением современных знаний в них. Например, на рисунке 11 видно, что BERT упоминается в статье The Word-in-Context Dataset for Evaluating Context-Sensitive Meaning Representations[34] еще до того, как про него вышла статья. В попытке понять как так про-

изошло и был выявлен данный факт. Собственно поэтому иногда возникали ситуации, когда разница в днях была отрицательной. Однако были и ситуации, когда на самом деле модель TrendSPECTER выделяла тренд раньше, чем он задокументирован в оценочном датасете, причем это была правильная дата (первая статья, использовавшая ключевую фразу). Но за разумное время проверить какие из случаев это редактирование, а какие правильное выделение не представляется возможным, поэтому в таблице все отрицательные значения обрезались в ноль. Проверка случаев представляется в качестве дальнейшей работы.

8 Заключение

В ходе работы были исследованы как классические, так и нейросетевые подходы к решению задачи выделения ключевых фраз и трендов. Была предложена новая модель для выделения ключевых фраз в научных документах, а также метод выделения трендов в них. Проведенные эксперименты показывают, что модель выделения ключевых фраз бесспорно лучше подходов к решению данной задачи, а метод выделения трендов не только находится на сравнимом с ними уровне, но и обладает множеством настраиваемых параметров, позволяющих решать задачу исходя из конкретных желаний пользователя.

Список литературы

[1] *Rolnick, David.* Tackling Climate Change with Machine Learning. — 2019.

[2] The impact of climate change on a cost-optimal highly renewable European electricity network / Markus Schlott, Alexander Kies, Tom Brown et al. // *Applied Energy.* — 2018. — nov. — Vol. 230. — Pp. 1645–1659.

[3] *et al., Martin Jinek.* A Programmable Dual-RNA–Guided DNA Endonuclease in Adaptive Bacterial Immunity / Martin Jinek et al. // *Science.* — 2012. — jun. — Vol. 337. — Pp. 816–821.

[4] *Hsu PD Lander ES, Zhang F.* Development and applications of CRISPR-Cas9 for genome engineering / Zhang F Hsu PD, Lander ES // *Cell.* — 2014. — jun. — Vol. 157(6). — Pp. 1262–1278.

[5] *Rose S., Engel D. Cramer N.* Automatic Keyword Extraction from Individual Documents / Engel D. Cramer N. Rose, S., W Cowley // *Text Mining.* — 2010. — mar. https://doi.org/10.1002/9780470689646.ch1.

[6] *Mihalcea, Rada.* TextRank: Bringing Order into Text / Rada Mihalcea, Paul Tarau // Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. — Barcelona, Spain: Association for Computational Linguistics, 2004. — . — Pp. 404–411. https://aclanthology.org/W04-3252.

[7] *Brin, Sergey.* The Anatomy of a Large-Scale Hypertextual Web Search Engine / Sergey Brin, Lawrence Page // *Computer Networks.* — 1998. — Vol. 30. — Pp. 107–117. http://www-db.stanford.edu/~backrub/google.html.

[8] *Grootendorst, Maarten.* KeyBERT: Minimal keyword extraction with BERT. — 2020. https://doi.org/10.5281/zenodo.4461265.

[9] *Vaswani, Ashish.* Attention Is All You Need. — 2017.

[10] *Devlin, Jacob.* BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. — 2019.

[11] *Han F., Magee C.L.* Testing the science/technology relationship by analysis of patent citations of scientific papers after decomposition of both science and technology / Magee C.L Han, F. // *Scientometrics*. — 2018. — Vol. 116. — Pp. 767–796. https://doi.org/10.1007/s11192-018-2774-y.

[12] *Patel, Sakshi.* A study of hierarchical clustering algorithms / Sakshi Patel, Shivani Sihmar, Aman Jatain // 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom). — 2015. — Pp. 537–541.

[13] *Newman, M. E. J.* Fast algorithm for detecting community structure in networks / M. E. J. Newman // *Physical Review E*. — 2004. — jun. — Vol. 69, no. 6.

[14] *Noack, Andreas.* Multi-level algorithms for modularity clustering. — 2008.

[15] *Grootendorst, Maarten.* BERTopic: Neural topic modeling with a class-based TF-IDF procedure / Maarten Grootendorst // *arXiv preprint arXiv:2203.05794*. — 2022.

[16] *Blei, David M.* Latent Dirichlet Allocation / David M. Blei, Andrew Y. Ng, Michael I. Jordan // *J. Mach. Learn. Res.* — 2003. — mar. — Vol. 3, no. null. — P. 993–1022.

[17] *Hoffman, Matthew.* Online Learning for Latent Dirichlet Allocation / Matthew Hoffman, Francis Bach, David Blei // Advances in Neural Information Processing Systems / Ed. by J. Lafferty, C. Williams, J. Shawe-Taylor et al. — Vol. 23. — Curran Associates, Inc., 2010. https://proceedings.neurips.cc/paper_files/paper/2010/file/71f6278d140af599e06ad9bf1ba03cb0-Paper.pdf.

[18] *Yadav, Vikas.* A Survey on Recent Advances in Named Entity Recognition from Deep Learning models / Vikas Yadav, Steven Bethard // Proceedings of the 27th International Conference on Computational Linguistics. — Santa Fe, New Mexico, USA: Association for Computational Linguistics, 2018. — . — Pp. 2145–2158. https://aclanthology.org/C18-1182.

[19] *Sari, Yunita.* Rule-based pattern extractor and named entity recognition: A hybrid approach / Yunita Sari, Mohd Fadzil Hassan, Norshuhani Zamin // 2010 International Symposium on Information Technology. — Vol. 2. — 2010. — Pp. 563–568.

[20] *Minh, Pham Quang Nhat.* A Feature-Based Model for Nested Named-Entity Recognition at VLSP-2018 NER Evaluation Campaign. — 2018.

[21] *Rabiner, L.* An introduction to hidden Markov models / L. Rabiner, B. Juang // *IEEE ASSP Magazine.* — 1986. — Vol. 3, no. 1. — Pp. 4–16.

[22] *Lafferty, John D.* Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data / John D. Lafferty, Andrew McCallum, Fernando C. N. Pereira // Proceedings of the Eighteenth International Conference on Machine Learning. — ICML '01. — San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001. — P. 282–289.

[23] *Hochreiter, Sepp.* Long short-term memory / Sepp Hochreiter, Jürgen Schmidhuber // *Neural computation.* — 1997. — Vol. 9, no. 8. — Pp. 1735–1780.

[24] *Schuster, M.* Bidirectional recurrent neural networks / M. Schuster, K.K. Paliwal // *IEEE Transactions on Signal Processing.* — 1997. — Vol. 45, no. 11. — Pp. 2673–2681.

[25] *Radford, Alec.* Improving Language Understanding by Generative Pre-Training / Alec Radford, Karthik Narasimhan. — 2018.

[26] *Liu, Yinhan.* RoBERTa: A Robustly Optimized BERT Pretraining Approach. — 2019.

[27] *Rumelhart D., Hinton G. Williams R.* Learning representations by back-propagating errors / Hinton G. Williams R. Rumelhart, D. // *Nature.* — 1986. — Vol. 323. — Pp. 533–536.

[28] *Zhang, Zhilu.* Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. — 2018.

[29] *Zhuang, Fuzhen.* A Comprehensive Survey on Transfer Learning. — 2020.

[30] *Cohan, Arman.* SPECTER: Document-level Representation Learning using Citation-informed Transformers. — 2020.

[31] SemEval-2010 Task 5: Automatic Keyphrase Extraction from Scientific Articles / Su Nam Kim, Olena Medelyan, Min-Yen Kan, Timothy Baldwin // Proceedings of the 5th International Workshop on Semantic Evaluation. — SemEval '10. — USA: Association for Computational Linguistics, 2010. — P. 21–26.

[32] *Loper, Edward.* NLTK: The Natural Language Toolkit. — 2002.

[33] *Hulth, Anette*. Improved Automatic Keyword Extraction given More Linguistic Knowledge / Anette Hulth // Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing. — EMNLP '03. — USA: Association for Computational Linguistics, 2003. — P. 216–223. https://doi.org/10.3115/1119355.1119383.

[34] *Pilehvar, Mohammad Taher*. WiC: the Word-in-Context Dataset for Evaluating Context-Sensitive Meaning Representations. — 2019.