Московский государственный университет имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математических методов прогнозирования

Карпинская Анна Викторовна

Автоматическое именование и суммаризация тем в вероятностном тематическом моделировании

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:
д.ф.-м.н., профессор РАН
Воронцов Константин Вячеславович

Москва, 2025

Содержание

1 Введение

Тематическое моделирование представляет собой одно из ключевых направлений современной обработки естественного языка (natural language processing, NLP), интенсивно развивающееся с конца 1990-х годов [5]. Под тематической моделью текстовой коллекции понимается вероятностное представление, описывающее, каким темам соответствуют отдельные документы, а также какие слова наиболее характерны для каждой темы. Такие модели относятся к области обучения без учителя (unsupervised learning), поскольку они строятся исключительно на основе статистических свойств текстов, без привлечения экспертной аннотации, тезаурусов или внешних баз знаний.

Вероятностные тематические модели (probabilistic topic models) описывают текстовую коллекцию как совокупность дискретных распределений: распределения тем в каждом документе и распределения слов в рамках каждой темы. Это позволяет формализовать тематику текстов в терминах латентных переменных, отражающих скрытую семантическую структуру данных.

Одним из наиболее значимых применений тематического моделирования является задача информационного поиска и разведочного анализа больших текстовых массивов (exploratory search). Методология нашла широкое применение в цифровых гуманитарных науках (digital humanities), филологии, культурологии, истории, социологии, политологии, журналистике и маркетинге [3, 6, 17]. В типичных сценариях тематическая модель позволяет получить представление о содержании коллекции без необходимости читать каждый документ, а также выделить релевантные фрагменты для последующего анализа. При этом исследователь может заранее не знать, какие именно темы окажутся значимыми, что делает тематическое моделирование особенно ценным инструментом для формулировки новых гипотез и выявления нетривиальных закономерностей.

Инструменты тематического моделирования обычно снабжаются пользовательским интерфейсом, позволяющим визуализировать темы в текстовой или графической форме [2]. Как правило, для каждой

темы выводится список слов, ранжированных по убыванию вероятностей в данной теме. Иногда он дополняется списком документов, также ранжированных по убыванию вероятностей в теме. Однако этой информации может оказаться недостаточно для понимания темы пользователем. Каждая тема должна уметь рассказать о себеї полно, точно, понятно для пользователя.

Задача автоматического именования темы (automatic topic labeling) состоит в том, чтобы подобрать для каждой темы лаконичный релевантный заголовок из заранее заготовленного списка фраз-кандидатов, проследив, чтобы разные темы не получили слишком похожие заголовки. Впервые эта задача была поставлена и решена в [12]. В последующих работах решение немного улучшалось, не меняясь концептуально [4, 7, 15]. Краткий заголовок темы полезен для навигации по списку тем и изучения модели, однако для понимания темы пользователем его также недостаточно.

Другой подход мог бы заключаться в автоматической суммаризации темы, когда генерируется краткое изложение или аннотация темы в виде связного текста. Задача суммаризации текстов имеет длинную предысторию и хорошо исследована, начиная с 50-х годов [**?**] и заканчивая современными обзорами [14, 19]. В частности, в 2019 году была опубликована статья [13], аннотация которой завершалась фразой sNote: The abstract above was not written by the authors, it was generated by one of the mo dels presented in this paper. Несмотря на успехи методов суммаризации и очевидную практическую востребованность, задача суммаризации отдельных тем в тематических моделях до сих пор никем не решалась, и даже не ставилась. Определённый шаг сделан в недавней работе [18], где ставится задача выделения тематических фраз, содержащих ключевые слова заданной темы. Поиск таких фраз несомненно полезен, но не является полноценной суммаризацией темы.

При этом есть немало исследований, в которых тематическое моделирование используется как вспомогательный инструмент на одном из этапов обработки данных для суммаризации коллекции документов (multi-document summarization), см. например [10, 16]. Идея заключается в том, чтобы с помощью тематической модели выделить

основные темы текстовой коллекции, затем из каждой темы отобрать наиболее репрезентативные фразы, и тем самым повысить полноту суммаризации. Очевидно, что используемые в этих работах методы поиска и выделения фраз по заданной теме могут быть применены также и для суммаризации темы. Серьёзной проблемой в области суммаризации текстов является оценивание качества решения. Общепринятым способом измерения качества уже более 20 лет остаётся метрика ROUGE [8, 9], хотя её недостатки давно осознаны и хорошо известны сообществу. Эта метрика позволяет сравнивать автоматическую суммаризацию с одной или несколькими суммаризациями, заранее написанными людьми. Сравнение основано на подсчёте совместно используемых слов или словосочетаний. При этом никак не оценивается связность, логичность, отсутствие фактических, речевых или стилистических ошибок.

При оценивании качества именования тем в [12] и последующих работах использовалось исключительно экспертное оценивание. К сожалению, такой подход не позволяет масштабировать экспериментальные исследования моделей. Для каждой модели именования (или суммаризации) приходится привлекать экспертов, что долго, дорого и субъекивно. Отдельной проблемой является создание масштабируемой меры качества, которую можно было бы один раз откалибровать, хотя бы для заданной текстовой коллекции, чтобы впоследствии по полученной размеченной выборке оценивать и сравнивать различные модели.

Настоящее исследование направлено на создание методов автоматического именования и суммаризации тем в вероятностных тематических моделях.

2 Постановка задачи

Дана большая коллекция документов D, для которой методом вероятностного тематического моделирования построена модель с заданным числом тем T. В частности, с помощью модели BigARTM [?] получены матрицы $\Phi = \{\phi_{wt}\}$ и $\Theta = \{\theta_{td}\}$, где:

- $\phi_{wt} = P(w \mid t)$ — распределение вероятностей слов w из словаря W по каждой теме $t \in T$,
- $\theta_{td} = P(t \mid d)$ — распределение тем по документам $d \in D$.

Матрица Φ характеризует содержание каждой темы через список наиболее характерных слов, а матрица Θ описывает тему через множество документов, в которых она представлена.

Задача: используя Φ, Θ и исходный корпус текстов, автоматически сгенерировать краткое текстовое описание для каждой темы. Формально, требуется построить отображение:

$$f : T \to L,$$

где L — множество текстовых меток, $l_t \in L$ — строка текста, оптимально характеризующая тему t.

Метка l_t должна быть:

- интерпретируемой (понятной человеку),
- информативной (отражающей содержание темы),
- уникальной в рамках набора тем.

3 Теоретическое обоснование

Тематическое моделирование – метод статистического анализа текстов, который выявляет скрытые в коллекции документы тематические структуры. В классических вероятностных моделях (PLSA, LDA) каждый документ d считается смесью тем, а каждая тема задаётся распределением по словам. Концептуально это сводится к матричному разложению: матрица частот слов-на-документы F приближённо раскладывается в произведение двух стохастических матриц Φ и Θ, где

$$\Phi = (\phi_{wt})$$
$$\Theta = (\theta_{td})$$

Элемент ϕ_{wt} задаёт условную вероятность слова w при фиксированной теме t ($\phi_{wt} = p(w|t)$), а θ_{td} — вероятность темы t в документе d ($\theta_{td} = p(t|d)$). В рамках PLSA/EM-алгоритма максимизируется правдоподобие модели, которое можно записать как

$$L(\Phi, \Theta) \;=\; \sum_{d \in D} \sum_{w \in d} n_{dw} \ln\Big(\sum_{t \in T} \phi_{wt}\, \theta_{td}\Big) \;\to\; \max_{\Phi,\Theta}$$

На этапе E оценки тем для каждого вхождения слова вычисляются по формуле Байеса:

$$p(t|d, w) \;=\; \frac{\phi_{wt}\, \theta_{td}}{\sum_{s \in T} \phi_{ws}\, \theta_{sd}}$$

Это означает, что сочетание матриц Φ и Θ позволяет определить апостериорное распределение тем для конкретного слова w в документе d. После этого на M-шаге EM-алгоритма матрицы Φ и Θ обновляются на основе подсчётов частотных ожиданий.

Модель LDA расширяет PLSA введением априорных распределений Дирихле на столбцы Φ и Θ. Практически это приводит к сглаживанию оценок параметров: после каждой итерации оценки пересчитываются по формулам

$$\phi_{wt} \;\propto\; n_{wt} + \beta_w, \qquad \theta_{td} \;\propto\; n_{td} + \alpha_t \;,$$

где n_{wt} и n_{td} – эмпирические частоты, а β_w, α_t – гиперпараметры Дирихле. Такие байесовские гиперпараметры контролируют разреженность или гладкость моделей. Однако в современных практических задачах часто возникает ситуация, когда реальная разреженность распределений тем не согласуется с предположениями Дирихле.

3.1 Тематическая модель BigARTM

Модель BigARTM (Additive Regularization of Topic Models) решает эту проблему, используя аддитивную регуляризацию, а не фиксированные априорные распределения. В ARTM обучение формулируется как многокритериальная оптимизация: максимизируется комбинированный функционал

$$L(\Phi, \Theta) + \sum_{i=1}^{n} \tau_i \, R_i(\Phi, \Theta) \;\; \to \max_{\Phi, \Theta},$$

где $L(\Phi, \Theta)$ – логарифм правдоподобия, а R_i – введённые регуляризаторы с весами τ_i. Такая формулировка позволяет гибко задавать различные требования к модели: например, можно явно включать регуляризаторы разреженности, сглаживания и декорреляции тем. Таким образом, BigARTM не пытается строить полностью генертивную модель текста, как LDA, а решает задачу оптимизации регуляризованного функционала. Это упрощает подбор модели под конкретные задачи: с помощью регуляризаторов можно, например, добиться более разреженных или, наоборот, более равномерных распределений тем, улучшая интерпретируемость результатов.

3.2 Локализация тем в тексте по Φ и Θ

После обучения модели BigARTM мы имеем матрицы Φ и Θ. Они содержат знания о том, какие слова характерны для каких тем и в

каких документах какие темы преобладают. Поэтому для каждого слова w на позиции i в документе d можно оценить вероятность того, что оно относится к теме t, исходя из данных модели. Используя формулу из EM-алгоритма, определим базовое распределение тем для данного слова:

$$p_0(t \mid w_i) \;=\; p(t \mid d, w_i) \;=\; \frac{\phi_{w_i t}\,\theta_{td}}{\sum_{s \in T} \phi_{w_i s}\,\theta_{sd}}\,.$$

Здесь $\phi_{w_i t} = p(w_i|t)$ – вероятность слова w_i в теме t, а $\theta_{td} = p(t|d)$ – доля темы t в документе d. Такое апостериорное распределение отражает, насколько слово w_i в данном документе «объясняется» каждой темой t. Именно поэтому матрицы Φ и Θ подходят для локализации тем в тексте: они позволяют вычислять $p(t|d, w)$ для каждого слова.

3.3 Экспоненциальное сглаживание $p(t|w, i)$

Однако каждая отдельная словоформа может принадлежать сразу к нескольким темам, и локальная семантика часто определяется контекстом. Чтобы учесть окружение слова, в методе вводится экспоненциальное скользящее среднее тем вокруг позиции i. Формально определим скорректированное распределение тем $p(t|w, i)$ как взвешенное среднее базовых распределений $p_0(t|w_j)$ соседних слов w_j в окне вокруг i с экспоненциальным затуханием веса по мере удаления:

$$\hat{p}(t|w, i) \;\propto\; \sum_{k=-K}^{K} \lambda^{|k|}\, p_0\big(t \mid w_{\,i+k}\big)\,, \qquad 0 < \lambda < 1,$$

где K – ширина окна, а λ – коэффициент затухания. В нормализованном виде это даёт

$$p(t|w, i) \;=\; \frac{\sum_{k=-K}^{K} \lambda^{|k|}\, p_0(t \mid w_{i+k})}{\sum_{k=-K}^{K} \lambda^{|k|}}\,.$$

Таким образом учтены взвешенные в обе стороны вклады контекстных слов: ближние по позиции слова дают больший вклад, далекие – экспоненциально меньший. Симметричное сглаживание обеспечивает справедливое учёт левого и правого контекста, в отличие от односторонних подходов, и соответствует идее, что значение слова определяется окружающими его терминами.

3.4 Кластеризация тематических фраз методом DBSCAN

После извлечения сглаженных тематических вероятностей $p(t|w,i)$ и построения списка ключевых фраз для каждой темы возникает задача устранения дублирующих и слабоинформативных выражений. Для этого применяется метод кластеризации фраз в семантическом пространстве. Каждая фраза представляется вектором с использованием предварительно обученной модели эмбеддингов, что позволяет переходить от текстов к точкам в $\mathbb{R}^d$.

Для группировки схожих по смыслу фраз используется алгоритм DBSCAN (Density-Based Spatial Clustering of Applications with Noise) — метод кластеризации на основе плотности. Он не требует указания числа кластеров и устойчив к шуму. Кластер определяется как связная область точек, в которой каждая точка имеет не менее minPts соседей в пределах радиуса ε.

Алгоритм DBSCAN опирается на следующие определения:

- ε-окрестность точки x — множество:

$$N_\varepsilon(x) = \{y \in \mathbb{R}^d \mid \|x - y\| \leq \varepsilon\},$$

- точка считается *ядровой*, если $|N\_\varepsilon(x)| \geq minPts$.

Алгоритм группирует все точки, доступные из ядровых по цепочке соседей, в один кластер. Остальные точки считаются шумом

и отбрасываются. Таким образом, метод автоматически отфильтровывает изолированные фразы, которые не разделяют семантику ни с одним кластером.

Применительно к задаче тематической аннотации, DBSCAN позволяет:

- выделить устойчивые формулировки, описывающие одну и ту же подтему;
- отбросить нерелевантные или слабо связанные по смыслу фразы;
- избежать необходимости заранее задавать число кластеров.

После кластеризации из каждого кластера выбирается несколько центральных фраз — ближайшие к среднему эмбеддингу, либо наиболее частотные в корпусе. Это приводит к финальному списку сжатых тематических формулировок, отражающих наиболее информативные аспекты каждой темы.

Таким образом, предложенный подход автоматически локализует темы в тексте через апостериорные вероятности из матриц Φ и Θ с учётом контекста. Экспоненциальное скользящее среднее позволяет получать «сглаженные» распределения $p(t|w,i)$, которые теоретически более устойчивы и семантически осмысленны, чем простые бесконтекстные оценки. Извлечённые фразы группируются в семантическом пространстве с помощью алгоритма DBSCAN, что позволяет отфильтровать шум и объединить схожие по смыслу выражения. В совокупности это обеспечивает более точную и интерпретируемую аннотацию тем по сравнению с традиционными методами.

4 Реализованный метод

4.1 Предварительная обработка данных

В рамках данного исследования использовался текстовый корпус данных с платформы Hugging Face под названием «Brawler/Medium-articles», содержащий статьи с ресурса Medium [20]. Прежде чем приступить к тематическому моделированию, были выполнены шаги предобработки текста. Весь корпус был очищен следующим образом: текст приводился к нижнему регистру, удалялись все символы пунктуации, а также устранены стоп-слова английского языка.

После очистки данных из корпуса отбирались только достаточно большие документы – были оставлены статьи длиной более 1500 и менее 16000 символов после удаления лишних элементов. Такой порог длины позволяет исключить короткие заметки и длинные рассказы и обеспечить, чтобы в модель поступали полноценные статьи, содержащие достаточно контента для выявления устойчивых тем.

Отфильтрованные и очищенные тексты были сохранены в формате, пригодном для тематического моделирования Vowpal Wabbit с указанием идентификатора документа и списка токенов, что послужило входными данными для обучения модели.

4.2 Подготовка корпуса данных

После выполнения всех шагов предобработки и фильтрации, описанных выше, в распоряжении остался корпус из 56407 текстов, представляющих собой полноразмерные статьи. На этом корпусе была обучена первая тематическая модель с использованием библиотеки BigARTM (Additive Regularization of Topic Models), поддерживающей обучение многотематических моделей с различными регуляризаторами, см. раздел 3.1.

Для построения модели было выбрано количество тем, равное 50. Это значение обеспечивало распределение примерно по тысяче

документов на каждую тему и позволяло модели охватить широкий спектр возможных тематик в корпусе. Обучение модели производилось с использованием регуляризаторов:

- Разреженность по документам (Sparse Θ)

  Использовался регуляризатор "SmoothSparseThetaRegularizer"с параметром $\tau = -0.05$. Он способствует тому, чтобы каждый документ был представлен небольшим числом тем, тем самым повышая тематическую чистоту.

- Разреженность по темам (Sparse Φ)

  Применялся регуляризатор "SmoothSparsePhiRegularizer"с $\tau = -0.1$, который заставляет темы концентрироваться на меньшем числе слов, отсекая фоновую лексику.

- Декорреляция тем

  Для повышения различимости тем был добавлен регуляризатор "DecorrelatorPhiRegularizer"с параметром $\tau = 10^3$. Он минимизирует перекрытие распределений слов между разными темами, что важно для интерпретируемости.

По результатам полученной матрицы распределений тем по документам $\Theta = (\theta_{td})$, были выделены документы, в которых значение θ_{td} по одной из тем превышало 0.8. В результате отбора был сформирован подкорпус из 932 документов, покрывающих 7 наиболее устойчивых и интерпретируемых тем. Для дальнейшего анализа была повторно обучена тематическая модель BigARTM на итоговых текстах.

4.3 Алгоритм аннотирования тем в тексте

После получения финального набора тем была разработана методика аннотирования текстов этими темами. Цель аннотирования – автоматически выделить в тексте ключевые фрагменты и фразы,

указывающие на ту или иную тему, на основе результатов тематического моделирования. Ниже подробно описан реализованный алгоритм аннотирования.

1. Распределение $p(t \mid w_i, i)$ с учётом контекста

Для оценки принадлежности слова w_i теме t используется позиционное распределение $p(t \mid w_i, i)$, которое учитывает контекст вблизи позиции i. Метод основан на экспоненциальном сглаживании базовых распределений тем по словам, подробнее изложенном в разделе 3.3. Смысл сглаживания — усилить вклад ближайших к w_i слов в оценку темы и ослабить влияние более удалённых.

Для каждой темы t вычисляется взвешенная сумма значений $\phi_t(w_j)$ — вероятностей слов в теме t, полученных из обученной модели:

$$S(t, i) = \sum_{j=i-L}^{i+L} \exp\left(-\frac{|i-j|}{\tau}\right) \phi_t(w_j),$$

где L — радиус окна, τ — коэффициент затухания. Эта сумма интерпретируется как "локальная плотность"темы t в окрестности слова w_i.

Для получения итогового распределения $p(t \mid w_i, i)$ выполняется нормализация по всем темам в данной позиции. Полученные вероятности используются на следующем этапе для выделения тематических фраз, указывающих на наиболее выраженные темы в тексте.

2. Выделение ключевых фраз на основе тематических плотностей

На следующем этапе полученное распределение тем по тексту используется для извлечения пояснительных фраз, которые будут служить аннотациями тем.

Алгоритм выделения фраз выглядит следующим образом:

1. Для каждой темы t из выбранного набора просматривается последовательность значений $p(t \mid w_i, i)$ по всем позициям i данного документа.

2. Из этой последовательности находятся наиболее высокие значения – те позиции в тексте, где вероятность темы t особенно велика относительно остальных.

3. Выбирается несколько топ-позиций для каждой темы. Каждая такая позиция i^* фактически соответствует конкретному слову w_{i^*}, которое в своем контексте наиболее явно указывает на тему t.

4. Вокруг каждой найденной позиции формируется сама ключевая фраза – небольшой отрывок текста, содержащий это слово и его ближайший контекст.

4.4 Кластеризация тематических фраз

Как отмечено в подразделе 3.4, после извлечения набора ключевых фраз необходимо устранить семантическую избыточность и удалить нерепрезентативные выражения. Для этого в пространстве эмбеддингов Sentence-BERT была выполнена кластеризация методом DBSCAN.

1. Каждая фраза преобразуется в вектор $x_i \in \mathbb{R}^d$.

2. Алгоритм DBSCAN объединяет фразы в кластеры C_k («ядро» плотности), если для вектора x_i выполняется

$$\left| \{x_j : \|x_i - x_j\|_2 \leq \varepsilon\} \right| \geq minPts.$$

3. Фразы, не вошедшие ни в один кластер, помечаются как шум и исключаются.

4. Для каждого кластера выбирается одна центральная фраза (наиболее близкая к медианному эмбеддингу), а все фразы, совпадающие по леммам с перестановкой слов, дополнительно схлопываются в единый вариант.

Итогом является компактный набор уникальных формулировок (по одной на кластер), отражающих основные подтемы внутри каждой темы. На рис. 1 и рис. 2 показаны двумерные UMAP-проекции фраз и разбиение DBSCAN для всех тем.

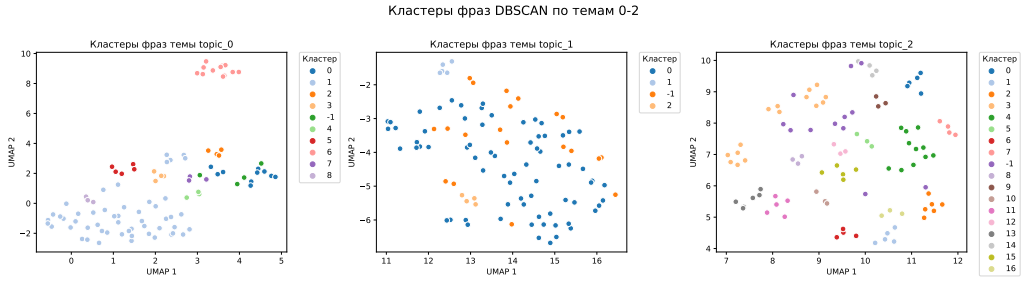

Рисунок 1: Результаты кластеризации DBSCAN: UMAP-проекции фраз для тем 0–2

Для последующей оценки качества аннотирования была вручную подготовлена эталонная разметка фраз, описывающих суть каждой темы. Для этого по результатам обученной модели были просмотрены топовые слова каждой из тем и отобраны документы с высокой тематической принадлежностью. На основе их содержания вручную выделялись фрагменты текста — фразы, которые наиболее точно и кратко описывают происходящее в пределах конкретной темы. Таким образом формировался эталонный набор фраз по каждой теме, пригодный для сравнения с результатами автоматического извлечения аннотаций.

Рисунок 2: Результаты кластеризации DBSCAN: UMAP-проекции фраз для тем 3–6

5 Экспериментальная оценка качества алгоритма

5.1 Метрики оценки качества

Для объективного сравнения различных настроек модели и алгоритма аннотирования был использован набор метрик, отражающих полноту тем, разнообразие словаря и равномерность распределения фраз. Ниже приведены определения и формулы каждой метрики.

Precision, Recall, F_1-measure

При сопоставлении предсказанных фраз с эталонными используются традиционные метрики бинарной классификации:

$$\text{Precision} = \frac{TP}{TP + FP}, \qquad \text{Recall} = \frac{TP}{TP + FN},$$

$$F_1 = \frac{2\,\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}},$$

где TP — число верно найденных фраз, FP — алгоритм выделил фразу, но в эталонной разметке её нет, FN — фраза есть в эталонной разметке, но не была найдена алгоритмом.

Coverage

Покрытие оценивает, какая доля документов имеет хотя бы одну корректно обнаруженную фразу:

$$\text{Coverage} = \frac{|\,\{d :\ \exists\,\text{фраза в } d\}|}{|D|}\,.$$

Вот более подробное описание метрики **JS-divergence**, которое ты можешь вставить как расширенное пояснение в подраздел с метриками (или как отдельный параграф):

—

JS-divergence

Метрика Jensen–Shannon divergence (JS-дивергенция) используется для оценки того, насколько равномерно распределены фразы между выбранными темами. Она измеряет «дистанцию» между фактическим распределением фраз по темам p и идеальным равномерным распределением u.

$$\text{JS}(p) = \frac{1}{2}\left[\text{KL}(p\|m) + \text{KL}(u\|m)\right], \qquad m = \frac{1}{2}(p + u)$$

где $\mathrm{KL}(a|b)$ — дивергенция Кульбака–Лейблера, вычисляемая как:

$$\mathrm{KL}(a\|b) = \sum_i a_i \log_2 \frac{a_i}{b_i}$$

Таким образом, JS-дивергенция сравнивает два распределения (p и u), учитывая их усреднённое распределение m как точку отсчёта. Чем ближе p к u, тем меньше $\mathrm{JS}(p)$.

Среднее число фраз

$$\overline{n}_{\mathrm{phr}} = \frac{1}{|D|} \sum_{d \in D} n_d,$$

где n_d — количество оставшихся фраз в документе d после всех стадий фильтрации.

BLEU-2 (n-gram)

Двух-граммовая версия BLEU вычисляется как

$$\text{BLEU-2} = \mathrm{BP} \cdot \exp\Big(\tfrac{1}{2}\big(\ln p_1 + \ln p_2\big)\Big),$$

$$p_n = \frac{\text{кол-во совпавших } n\text{-грамм}}{\text{кол-во } n\text{-грамм в предсказании}},$$

где BP — штраф за длину (brevity penalty).

ROUGE-L

ROUGE-L основан на длине наибольшей общей подпоследовательности (LCS) между эталоном и предсказанием:

$$\text{ROUGE-L} = \frac{(1+\beta^2)\, P_{\mathrm{LCS}}\, R_{\mathrm{LCS}}}{R_{\mathrm{LCS}} + \beta^2 P_{\mathrm{LCS}}}, \qquad \beta = \frac{P_{\mathrm{LCS}}}{R_{\mathrm{LCS}}},$$

где $P_{\mathrm{LCS}} = \frac{\mathrm{LCS}}{\mathrm{len(pred)}}$ и $R_{\mathrm{LCS}} = \frac{\mathrm{LCS}}{\mathrm{len(gold)}}$.

Семантические Precision / Recall / F_1

На эмбеддингах Sentence-BERT вычисляется матрица похожести $S_{ij} = \cos(x_i^{\text{gold}}, x_j^{\text{pred}})$. Тогда

$$\text{SemPrecision} = \frac{1}{|P|}\sum_j \max_i S_{ij}, \qquad \text{SemRecall} = \frac{1}{|G|}\sum_i \max_j S_{ij},$$

$$\text{Sem}F_1 = \frac{2 \cdot \text{SemPrecision} \cdot \text{SemRecall}}{\text{SemPrecision} + \text{SemRecall}}.$$

Redundancy

Избыточность предсказанного набора оценивается как доля повторяющихся токенов:

$$\text{Redundancy} = 1 - \frac{\left|\text{uniq}(\text{tokens}(\text{pred}))\right|}{\left|\text{all tokens}(\text{pred})\right|}.$$

Значение 0 означает отсутствие повторов, 1 — полную однотипность словаря.

5.2 Подбор гиперпараметров порога τ и размера окна

Эксперимент направлен на изучение влияния двух гиперпараметров алгоритма — порога тематической значимости τ и ширины контекстного окна — на характеристики извлекаемых фраз. Порог τ определяет жёсткость фильтрации позиций по теме, а размер окна регулирует объём учёта контекста при сглаживании (см. Раздел 3.3). Параметры варьировались по сетке

$$\tau \in \{2,\, 7,\, 10\}, \qquad \text{window} \in \{4,\, 6,\, 12\}\,\text{слов}.$$

Каждая из девяти комбинаций была проверена на одном и том же наборе коллекций.

Полнота

На рис. 3 максимальное значение метрики recall (см. раздел 5.1) 0.854 достигается при $\tau = 2,\ w = 4$. При увеличении окна полнота почти не растёт (примерно 0.72–0.76 для остальных конфигураций), а при увеличении порога τ заметно снижается.

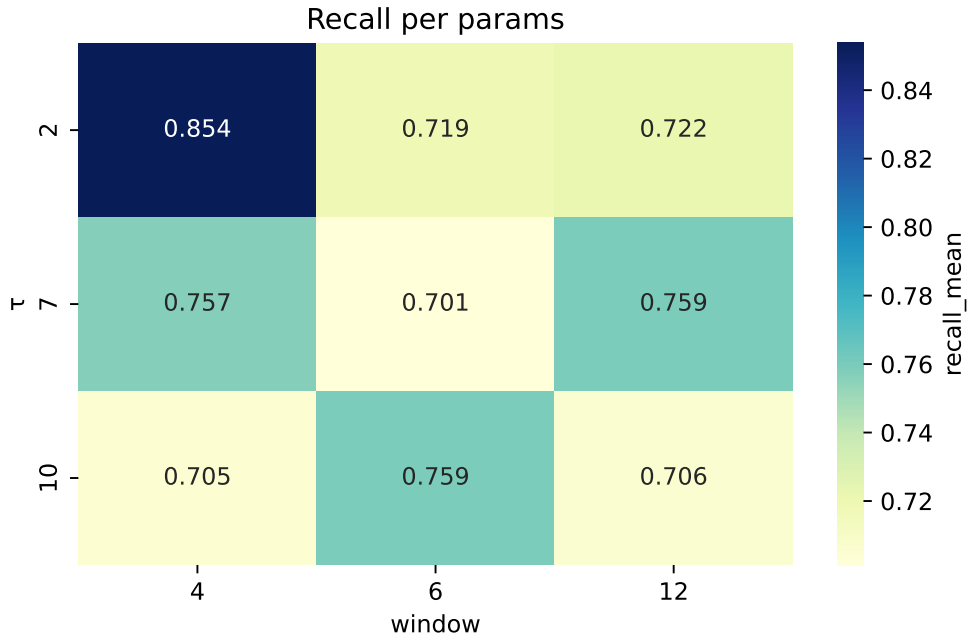

Рисунок 3: Полнота аннотаций (Recall) для различных значений τ и окна

Таким образом, низкий порог критически важен для полного покрытия эталонных фраз; окно оказывает второстепенное влияние.

Объём вывода

Из рис. 4 видно, что среднее количество фраз (см. раздел 5.1) монотонно растёт как с увеличением порога, так и с расширением окна — от 1340 фраз ($\tau = 2,\ w = 4$) до 3192 фраз ($\tau = 10,\ w = 12$). Причина — более широкий контекст даёт длинные выражения, а

высокий порог пропускает только распространённые и потому часто встречающиеся фразы.

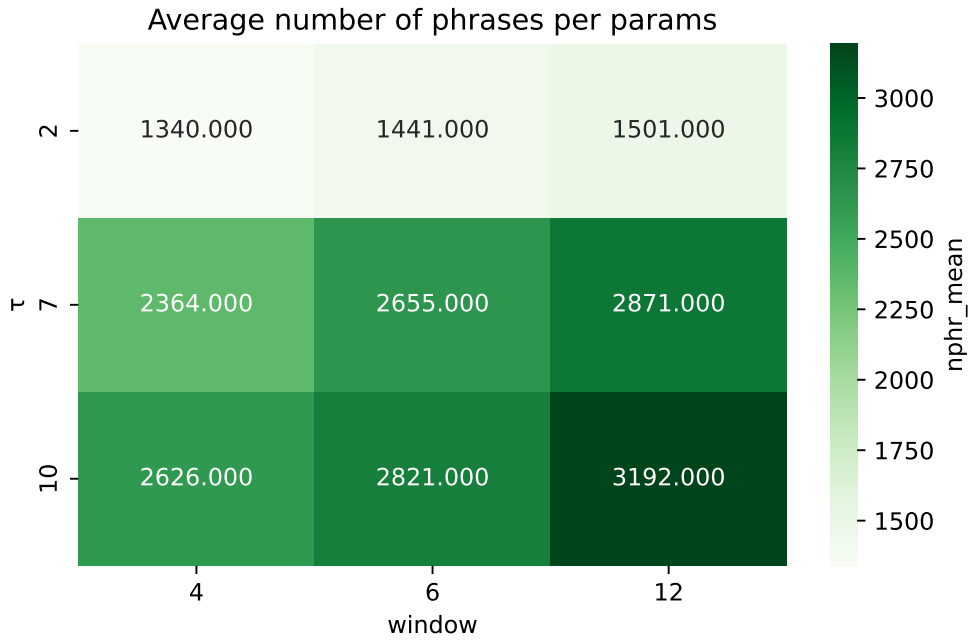

Рисунок 4: Среднее число извлечённых фраз

Равномерность тем

JS-дивергенция (см. раздел 5.1) на рис. 5 меньше всего при $\tau = 7$, $w = 12$ (0.001), что означает почти равное распределение фраз по семи темам. Наиболее неравномерный вариант — $\tau = 2$, $w = 4$ (JS $= 0.073$): большая часть фраз концентрируется в нескольких «легко извлекаемых» темах.

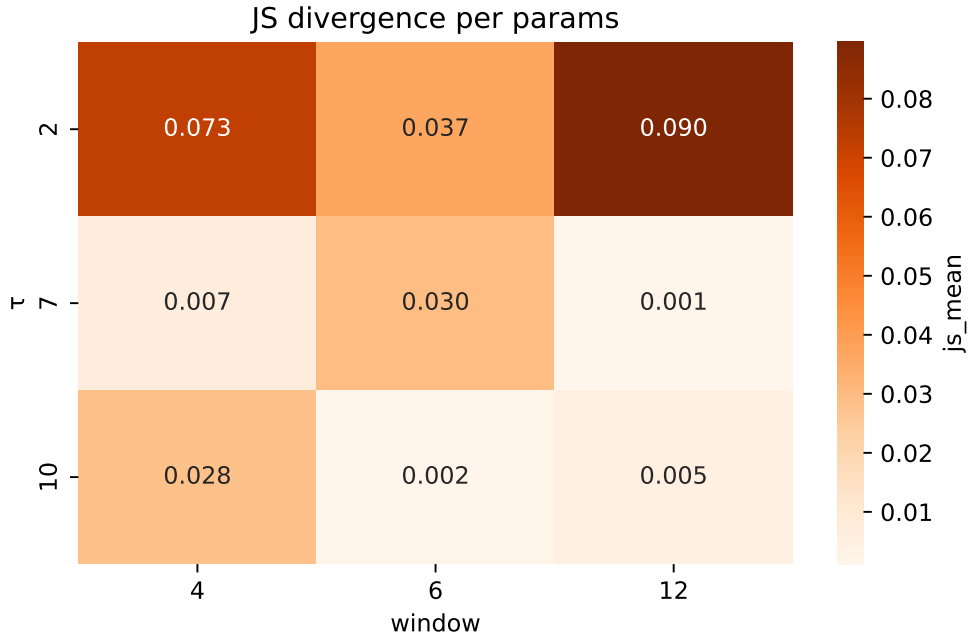

Рисунок 5: JS-дивергенция распределения фраз по темам

В итоге, порог τ оказывает наибольшее влияние: низкие значения обеспечивают максимальный recall, но уменьшают равномерность и слегка снижают разнообразие. Ширина окна регулирует соотношение «мелких» и «расширенных» фраз, слабо влияя на полноту, но повышая объём вывода. В данных условиях выбирается $\tau = 2$, $w = 6$ как компромисс: достаточный recall (0.719) при относительно высоком разнообразии (0.168) и умеренном объёме результата.

5.3 Оценка времени и потребления памяти при увеличении размера корпуса

Эксперимент нацеливался на проверку масштабируемости алгоритма при росте числа документов. Анализировались три показателя:

- общее число извлечённых фраз;

- время выполнения;
- пиковое потребление оперативной памяти.

Алгоритм запускался на подпоследовательностях корпуса размером $N \in \{10, 25, 50, 100, 200, 500\}$ документов (параметры неизменны: окно $w = 6$, порог $\tau = 2$).

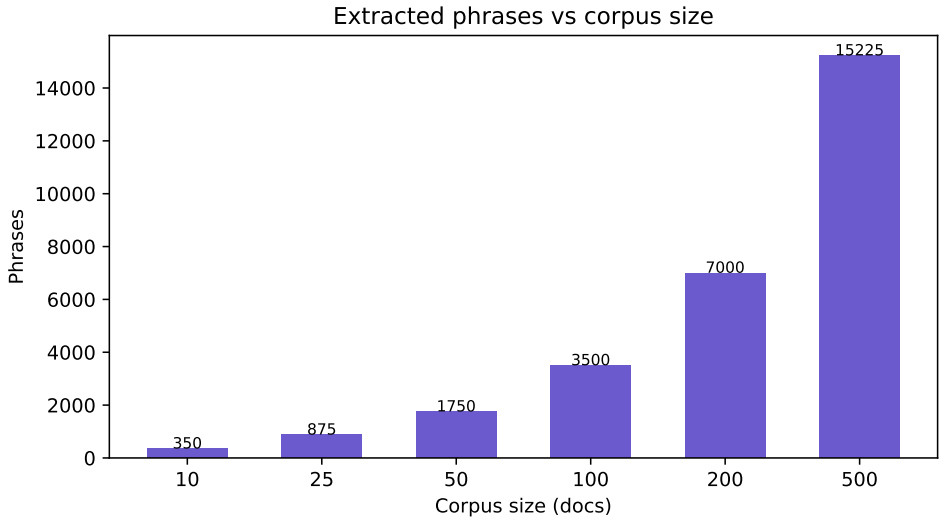

Рисунок 6: Количество извлечённых фраз в зависимости от размера корпуса

Число фраз. Как видно из рис. 6, зависимость строго линейна: от 350 фраз при 10 документах до 15 225 фраз при 500. Средняя отдача практически постоянна и составляет $\approx 30-35$ фраз на документ. Рост корпуса не приводит к «взрывному» появлению дублирующихся фраз, что подтверждает независимую обработку каждого текста.

Время. На рис. 7 кривая времени почти линейна: 4.2 с для 10 документов и 124.6 с для 500. Отношение «секунд на документ» стабилизируется вокруг 0.25–0.30 с. Таким образом, алгоритм демонстрирует сложность $O(N)$ по времени.

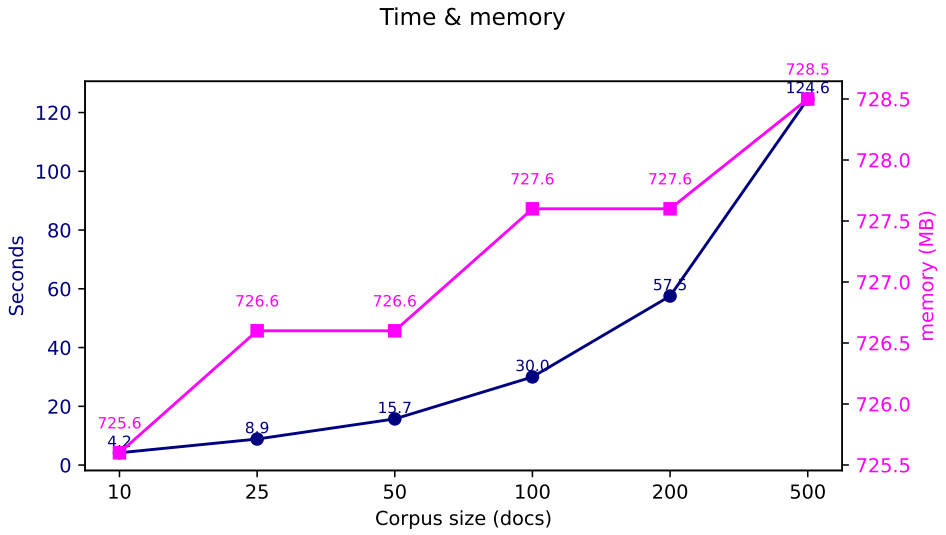

Рисунок 7: Время выполнения и пиковое потребление памяти

Память. Пиковое потребление RAM остаётся практически постоянных: от 725.5 МБ при 10 документах до 728.5 МБ при 500. Рост менее 3 МБ на порядок увеличения данных указывает, что основная память расходуется на модели и вспомогательные структуры, а не на сами тексты.

Предложенный алгоритм линейно масштабируется по времени и практически не зависит от размера корпуса по памяти, что позволяет обрабатывать большие коллекции без пропорционального роста аппаратных требований.

5.4 Сравнение с эталонной разметкой

Заключительный эксперимент сопоставляет фразы, извлечённые алгоритмом, с ручной разметкой эксперта. Для каждой темы вычислялись метрики (см.раздел 5.1): BLEU-2, ROUGE-L, семантический F_1 (SentenceTransformers + cosine) и Coverage. Результаты сведены в теплокарту (рис. 8).

Для «политических» тем 0 и 5 семантический F_1 достигает 0.47 и

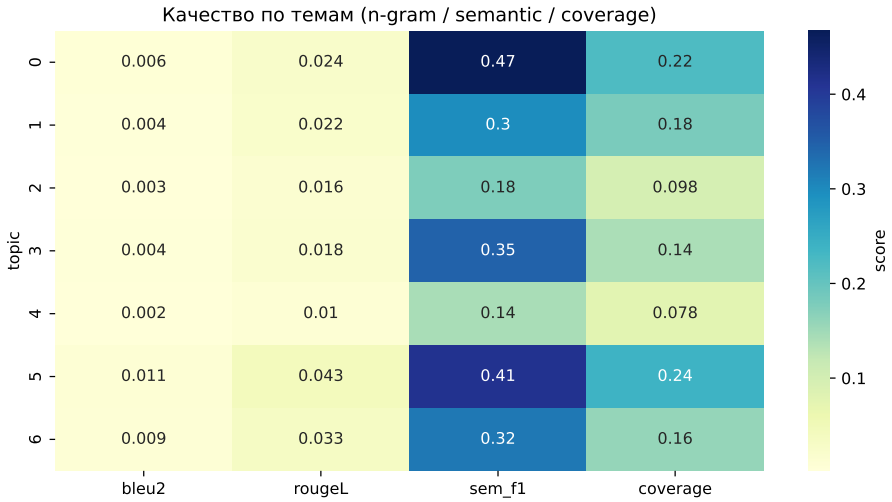

Рисунок 8: Качество по темам: n-gram, семантика и покрытие

0.41, тогда как темы 2 и 4 демонстрируют минимальное совпадение ($F_1 \approx 0.14$). Низкие значения BLEU/ROUGE (не выше 0.04) подтверждают, что формальные совпадения строк редки, а оценку следует делать по смыслу. Наиболее равномерными по покрытию оказались темы 0 и 5 (примерно 24 % документа содержат хотя бы одну верную фразу), худшее покрытие — у темы 4 (7.8 %).

Примеры соответствий (тема 6 - "Машинное обучение и Data Science").

Как видно из табл. 1, алгоритм корректно схватывает ключевые понятия (*ensemble learning*, *hyperparameter tuning*), однако пропускает узкие, но важные фразы о *data leakage*. Это отражается в семантическом recall 0.324 и относительно низком покрытии (16 % документов).

Эталонная фраза	**Фраза алгоритма**
different machine learning algorithms searching different trends patterns ...	evaluation metrics essential understanding performance machine learning models ...
ensemble learning inherently increase robustness reduce overfitting ...	ensemble learning inherently increase robustness reduce overfitting combine algorithms ... ✓
important tune hyperparameters right search method grid search random search ...	important tune hyperparameters right search method grid search random search bayesian optimization ... ✓
deep learning models require large training data computational resources ...	deep learning models require large training data computational resources overkill smaller structured tabular datasets ... ✓

Таблица 1: Совпадения и пропуски для темы 6

Для повышения recall и coverage планируется:

1. расширить окно контекста с 6 до 8 слов (см. результаты подбора гиперпараметров в разделе 5.2);

2. добавить синонимический словарь для детекции терминов;

Таким образом, сравнение с эталоном показывает, что алгоритм уже демонстрирует приемлемую точность, но требует доработки для повышения полноты — особенно в технически насыщенных темах вроде «Машинное обучение».

6 Заключение

В данной работе предложен и исследован алгоритм автоматической аннотации тем, основанный на вероятностном тематическом моделировании. Разработанный метод позволяет выделять фразовые конструкции, репрезентирующие смысл тем, с учётом локального контекста слова в документе. Это существенно повышает интерпретируемость тематических моделей, позволяя перейти от абстрактных распределений слов к конкретным поясняющим фразам.

Ключевая идея метода состоит в том, чтобы оценивать для каждого слова его тематическую значимость с опорой на окрестный контекст. Для этого использовано сглаженное распределение апостериорных вероятностей $p(t \mid w, i)$, полученное на основе матриц Φ и Θ тематической модели. В качестве ядра сглаживания применено симметричное экспоненциальное среднее по окну фиксированной ширины. Параметры сглаживания (ширина окна и коэффициент затухания) варьировались в ходе экспериментов, что позволило выявить наиболее устойчивые конфигурации.

Для отбора значимых позиций в тексте использовались локальные максимумы тематических плотностей по каждой теме. Окрестности этих позиций конвертировались в фразы — последовательности слов фиксированной длины. Далее применялась семантическая кластеризация с использованием эмбеддингов Sentence-BERT и алгоритма DBSCAN, что позволило устранить шумовые фразы и объединить перефразировки. После кластеризации выполнялось схлопывание схожих формулировок в пределах одного кластера: сохранялась центральная фраза, а близкие по составу и порядку слова удалялись как дубликаты. Этот этап значительно снизил избыточность и повысил компактность результата.

Экспериментальная оценка включала подбор гиперпараметров, сравнение с эталонной разметкой и проверку масштабируемости алгоритма. Выявлены сильные и слабые стороны предложенного подхода. Алгоритм успешно формирует содержательные аннотации в

случае чётко выраженной тематической структуры документа, демонстрируя средний семантический $F_1 = 0.47$ на всех темах. В частности, на технической теме «Machine Learning» достигнуты значения SemPrecision = 0.319 и SemRecall = 0.324. Однако в темах со слабо выраженными границами наблюдаются потери полноты, связанные с переотсевом редких и синонимичных формулировок. Проведённый анализ также показал, что качество аннотирования чувствительно к выбору параметров: порог тематической значимости и ширина окна существенно влияют на полноту, разнообразие и точность.

С точки зрения вычислительной эффективности метод показал хорошее масштабирование: время обработки растёт линейно по числу документов, а использование оперативной памяти почти не изменяется с увеличением корпуса. Это позволяет применять алгоритм к большим коллекциям текстов без существенного роста ресурсных затрат.

Полученные результаты открывают ряд направлений для дальнейшего развития. Перспективными являются:

- улучшение семантического покрытия за счёт интеграции внешних словарей синонимов и перефразов;
- автоматическая настройка параметров сглаживания в зависимости от свойств темы;
- внедрение механизмов тематической сегментации текста для повышения локальности фраз;
- оптимизация структуры хранения и расчёта вероятностей для работы с многомиллионными корпусами.

Таким образом, предложенный подход объединяет интерпретируемость, автоматичность и вычислительную эффективность, что делает его полезным инструментом в задачах тематического анализа и построения контекстных аннотаций на реальных текстовых данных.

Список литературы

[1] Vorontsov K.V., Potapenko A.A. Additive Regularization of Topic Models // Machine Learning. – 2015. – Т. 101. – № 1–2. – С. 303–323.

[2] Айсина Р. М. Обзор средств визуализации тематических моделей коллекций тек- стовых документов // Машинное обучение и анализ данных (http://jmlda.org) . 2015. Т. 1, 11. С. 15841618.

[3] Boyd-Graber J., Hu Y., Mimno D. Applications of topic mo dels // Foundations and Trends in Information Retrieval . 2017. Vol. 11, no. 2-3. Pp. 143296.

[4] Gourru A., Velcin J., Roche M., Gravier C., Poncelet P. United we stand: Using multiple strategies for topic lab eling // 23rd International Conference on Applications of Natural Language to Information Systems, NLDB 2018, Paris, France, June 13-15, 2018. Berlin, Heidelb erg: Springer-Verlag, 2018. P. 352363.

[5] Hofmann T. Probabilistic latent semantic indexing // Pro ceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, USA: ACM, 1999. Pp. 5057.

[6] Jelodar H., Wang Y., Yuan C., Feng X., Jiang X., Li Y., Zhao L. Latent dirichlet allo cation (LDA) and topic modeling: mo dels, applications, a survey // Multimedia Tools and Applications . 2019. Vol. 78, no. 11. Pp. 1516915211.

[7] Kinariwala S. A., Deshmukh S. Onto TML: Auto-labeling of topic models // Journal of Integrated Science and Technology . 2021. Vol. 9, no. 2. Pp. 85-91.

[8] Lin C.-Y. ROUGE: A package for automatic evaluation of summaries // Text Summarization Branches Out. Barcelona, Spain: Asso ciation for Computational Linguistics, 2004. Pp. 7481.

[9] Lin C.-Y., Cao G., Gao J., Nie J.-Y. An information-theoretic approach to automatic evaluation of summaries // Pro ceedings of the Human Language Technology Conference of the NAACL, Main Conference. New York City, USA: Asso ciation for Computational Linguistics, 2006. Pp. 463470.

[10] Litvak M., Vanetik N., Liu C., Xiao L., Savas O. Improving summarization quality with topic mo deling // Pro ceedings of the 2015 Workshop on Topic Mo dels: Post- Pro cessing and Applications. New York, NY, USA: Asso ciation for Computing Machinery, 2015. Pp. 3947.

[11] Luhn H. P. The automatic creation of literature abstracts // IBM Journal of Research and Development . 1958. Vol. 2, no. 2. Pp. 159165.

[12] Mei Q., Shen X., Zhai C. Automatic lab eling of multinomial topic mo dels // Pro ceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA: Asso ciation for Computing Machinery, 2007. Pp. 490499.

[13] Subramanian S., Li R., Pilault J., Pal C. On extractive and abstractive neural do cument summarization with transformer language mo dels. 2019. 3

[14] Torres-Moreno J.-M. Automatic Text Summarization. John Wiley Sons, 2014. 320 pp.

[15] Truica C.-O., Apostol E.-S. Tlatr: Automatic topic lab eling using automatic (domain-sp ecic) term recognition // IEEE Access . 2021. Vol. 9. Pp. 76624 76641.

[16] Wang D., Zhu S., Li T., Gong Y. Multi-do cument summarization using sentence- based topic mo dels // Pro ceedings of the ACL-IJCNLP 2009 Conference Short Pap ers. Suntec, Singap ore: Asso ciation for Computational Linguistics, 2009. Pp. 297300.

[17] Wesslen R. Computer-assisted text analysis for so cial science: Topic mo dels and b eyond // CoRR . 2018. Vol. abs/1803.11045.

[18] Wil liams L., Anthi E., Arman L., Burnap P. Topic mo delling: Going b eyond token outputs // Big Data and Cognitive Computing . 2024. Vol. 8, no. 5. P. 44.

[19] Zhang Y., Jin H., Meng D., Wang J., Tan J. A comprehensive survey on pro cess- oriented automatic text summarization with exploration of llm-based metho ds. 2025. 4

[20] Brawler. “Medium-articles” Dataset. – HuggingFace, 2023.