

МЕТОДЫ ПСЕВДОРАЗМЕТКИ ДЛЯ АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ КЛЮЧЕВЫХ СЛОВ НА РУССКОМ ЯЗЫКЕ

**Яушев Камиль (Yaushev Kamil)^{a,*}, Лукашевич Наталья Валентиновна (Loukachevitch
Natalia Valentinovna)^{b,**}**

^a Московский государственный технический университет им. Н.Э. Баумана, 105005,
Россия, Москва, 2-я Бауманская ул., д. 5, стр. 1

^b Московский государственный университет имени М.В.Ломоносова, 119991,
Россия, Москва, Ленинские горы, д. 1

** e-mail: kyaush@mail.ru*

*** e-mail: louk_nat@mail.ru*

Аннотация

В статье исследуются методы псевдоразметки в задаче автоматической генерации ключевых слов для русскоязычных научных статей в условиях дефицита размеченных данных. Основное внимание уделяется автоматическому формированию обучающих данных на основе неразмеченного корпуса. Анализируются два взаимодополняющих подхода к построению псевдоразметки. Локальная схема использует заголовок, аннотацию и автоматически выделенные терминологические единицы текущего документа; часть этих единиц маскируется, после чего генеративная модель обучается восстанавливать содержательные фразы. Глобальная схема опирается на внешнюю коллекцию научных текстов и методы семантического поиска, что позволяет расширять обучающий материал формулировками, близкими содержанию документа, в том числе не представленными в аннотации в явном виде. Исходным этапом построения псевдоразметки служит статистико-лингвистическая фильтрация фраз, выделенных

статистическими методами. Результирующая выборка используется при формировании псевдометок и выборе маскируемых фрагментов для обучения генеративных моделей, а также при построении положительных и отрицательных примеров для контрастивного обучения и последующего семантического ранжирования. Экспериментальное сравнение показывает, что локальная схема псевдоразметки с маскированием даёт наилучшие результаты среди конфигураций на базе модели `gu-mbart-summ`, тогда как глобальная схема в текущей реализации не обеспечивает улучшения по сравнению с локальной. При этом построение контрастивных примеров на основе псевдоразметки повышает качество семантического ранжирования, а предварительное обучение на локальной псевдоразметке позволяет существенно сократить отставание компактной модели от более крупного генеративного базового уровня.

Ключевые слова: генерация ключевых слов, автоматическая разметка, маскирование, контрастивное обучение, фильтрация ключевых слов.

Введение

Ключевые слова служат одним из основных средств индексирования научных публикаций, тематической навигации и информационного поиска, поскольку позволяют в сжатой форме зафиксировать предметную область документа, его основные понятия и смысловые акценты. В задачах автоматической обработки научных текстов они рассматриваются как компактная форма представления содержания документа, упрощающая последующую тематическую классификацию, поиск релевантных материалов и организацию доступа к научной информации [8].

Автоматическая генерация ключевых слов по русскоязычным научным аннотациям представляет собой сложную задачу. Это связано как с богатой морфологией русского языка и вариативностью научных формулировок, так и с тем, что значительная часть релевантных ключевых слов не представлена в аннотации в явном виде, а восстанавливается лишь на уровне смысловой интерпретации текста [8, 9]. Для русскоязычных научных аннотаций показано, что доля таких единиц может быть высокой, поэтому методы, ориентированные исключительно на выделение фраз, явно представленных в тексте, не обеспечивают полного покрытия множества содержательно значимых ключевых слов [8].

В связи с этим особый интерес представляют генеративные модели, основанные на трансформерной архитектуре [19], поскольку они ориентированы не только на выявление уже присутствующих в тексте единиц, но и на восстановление более глубоких смысловых соответствий. В отличие от чисто экстрактивных подходов, такие модели способны учитывать смысловые связи внутри документа и порождать ключевые слова, которые не имеют буквального выражения в исходном тексте, но соответствуют его содержанию на концептуальном уровне. Для русскоязычных научных аннотаций уже показано, что дообученные модели mT5 и mBART в ряде случаев превосходят классические методы выделения ключевых слов в пределах домена [8]. Вместе с тем их применение существенно ограничивается дефицитом размеченных данных, необходимых для устойчивого обучения, что особенно заметно в условиях ограниченного объема специализированных русскоязычных корпусов [9].

В условиях недостатка ручной разметки особое значение приобретают подходы, позволяющие извлекать обучающие данные из неразмеченных корпусов [10]. Однако

применительно к русскоязычным научным аннотациям по-прежнему остаётся недостаточно изученным вопрос о том, каким образом формировать такую псевдоразметку, которая не только снижала бы влияние формального и семантического шума, но и расширяла бы множество релевантных ключевых слов, включая единицы, отсутствующие в тексте в явном виде, но существенные для передачи его содержания. Тем самым на первый план выходит задача построения такой схемы обучения, в которой автоматически формируемые метки могли бы использоваться не как вспомогательное дополнение, а как систематический источник информации для различных этапов обработки.

В статье рассматривается многоэтапный подход к автоматической генерации ключевых слов на основе русскоязычных научных аннотаций, при котором статистико-лингвистическая обработка автоматически выделенных фраз лежит в основе двух взаимодополняющих схем псевдоразметки, предназначенных для формирования обучающих данных разного типа. Локальная схема опирается на текущий документ и используется при предварительном обучении генеративной модели в режиме восстановления содержательных фрагментов. Глобальная схема, в свою очередь, опирается на внешнюю коллекцию научных текстов и семантический поиск, что позволяет расширять обучающие данные формулировками, содержательно близкими документу, но буквально в нём не представленными. Тем самым предлагаемый подход объединяет генерацию ключевых слов, выделение единиц, явно представленных в тексте, и последующее семантическое ранжирование кандидатов в рамках общей процедуры, направленной на более полное и устойчивое восстановление ключевой информации документа.

1. Близкие работы

В исследованиях по автоматическому выделению и генерации ключевых слов обычно выделяют два основных класса подходов, различающихся как по характеру формируемого результата, так и по способу соотнесения этого результата с исходным текстом документа. К первому классу относятся экстрактивные методы, извлекающие значимые слова и словосочетания непосредственно из текста. К ним принадлежат статистические методы, основанные на частотных характеристиках и векторном представлении документа, в частности TF-IDF [17] и YAKE! [3], а также графовые методы ранжирования, такие как TextRank [15] и TopicRank [2]. Достоинствами этих подходов являются относительная простота, интерпретируемость и независимость от размеченных данных. Вместе с тем они, как правило, ограничены единицами, явно представленными в документе, и потому не позволяют в полном объёме решать задачу порождения ключевых слов, отсутствующих в тексте в явном виде, но существенных для адекватного отражения его содержания.

Дальнейшее развитие этого направления связано с распространением нейросетевых моделей, способных учитывать более сложные смысловые зависимости внутри документа и выходить за пределы поверхностного совпадения между текстом и целевыми ключевыми единицами. Ключевую роль в этом переходе сыграли предварительно обученные модели разных архитектур: модели энкодерного типа, в частности BERT [5], а также модели типа «энкодер—декодер», прежде всего T5 [16] и mBART [18]. Появление таких моделей сделало возможной генеративную постановку задачи, в которой система не только извлекает уже имеющиеся в тексте словосочетания, но и порождает содержательно релевантные формулировки, соответствующие документу на уровне смысловой

интерпретации. Именно поэтому абстрактные подходы стали естественной основой для задач, в которых требуется восстанавливать ключевые слова, не представленные в тексте буквально.

Для русскоязычных научных текстов данное направление также уже получило развитие. В работе [8] исследовано применение моделей mT5 и mBART к генерации ключевых слов по корпусам русскоязычных научных аннотаций из нескольких предметных областей; показано, что такие модели в пределах домена могут превосходить базовые методы выделения ключевых слов по качеству получаемых результатов. В последующих исследованиях для русскоязычных научных аннотаций рассматривались и более крупные генеративные модели, что дополнительно подтвердило перспективность генеративной постановки задачи, но одновременно показало, что богатая морфология русского языка, вариативность научных формулировок и ограниченный объем доступных обучающих данных по-прежнему затрудняют построение устойчивых решений [9]. Тем самым в литературе уже обозначено основное противоречие данной задачи, состоящее в том, что необходимость моделирования сложных смысловых связей сочетается с дефицитом качественных обучающих данных.

В условиях ограниченной ручной разметки особый интерес представляют методы, позволяющие автоматически формировать обучающие данные на основе неразмеченных корпусов. В работе [10] показано, что даже относительно простой источник псевдоразметки, основанный на использовании заголовка документа, может быть полезен для обучения моделей генерации ключевых слов, особенно в малоресурсной постановке и в случаях, когда требуется восстанавливать ключевые единицы, отсутствующие в тексте в явном виде. Другое важное направление связано с объединением генерации, выделения

значимых фрагментов и их последующего семантического ранжирования в рамках одной системы. В работе [4] предложен подход, сочетающий модель, совмещающую выделение и генерацию ключевых слов, контрастивное обучение представлений фраз и отдельный этап семантического ранжирования. Полученные результаты показывают, что автоматическое построение меток, контрастивное обучение и последующий отбор кандидатов целесообразно рассматривать не как изолированные стратегии, а как взаимодополняющие компоненты современных систем генерации ключевых слов, усиливающие друг друга на разных этапах обработки.

Рассматриваемый в статье подход соотносится с современным направлением исследований в области генеративно-контрастивного обучения применительно к задаче генерации ключевых слов, включая работы, в которых сочетаются автоматическое формирование обучающих данных, генерация кандидатов и их последующее семантическое ранжирование [4]. В предлагаемой работе эта общая схема адаптируется к русскоязычным научным аннотациям и дополняется собственными механизмами локальной и глобальной псевдоразметки, статистико-лингвистической фильтрации и отдельным этапом семантического ранжирования кандидатов.

2. Постановка задачи

Рассматривается документ, заданный заголовком и аннотацией. Требуется сформировать набор ключевых слов, отражающих содержание статьи. При применении системы на вход статистических алгоритмов и генеративных моделей подаётся только текст аннотации, тогда как заголовок используется лишь при построении обучающих данных на неразмеченном корпусе. Такое разделение сохраняет прикладную постановку задачи и

одновременно позволяет использовать дополнительную информацию заголовка на этапе предварительного обучения.

С точки зрения соотнесения с аннотацией различаются два типа ключевых слов. К первому относятся единицы, явно представленные в тексте: после лемматизации они совпадают с непрерывными фрагментами аннотации. Ко второму относятся ключевые слова, не представленные в тексте в явном виде: они не имеют буквального вхождения в аннотацию, но семантически соответствуют её содержанию. Это различие существенно как для построения моделей, так и для последующей оценки результатов, поскольку экстрактивные методы ориентированы прежде всего на выделение явно присутствующих фраз, тогда как абстрактивные подходы позволяют порождать ключевые слова, отсутствующие в тексте, но релевантные его содержанию.

Для решения поставленной задачи используются две коллекции документов. Первая представляет собой размеченный корпус, который применяется для дообучения моделей и итогового тестирования. Вторая является крупным неразмеченным корпусом, используемым при построении псевдоразметки; для документов этой коллекции, помимо аннотации, дополнительно доступен заголовок. Тем самым размеченный корпус служит источником эталонных ключевых слов, тогда как неразмеченный корпус используется для автоматического формирования дополнительных обучающих данных.

Во всех рассматриваемых конфигурациях на вход моделей и статистических методов подаётся исходный текст аннотации без предварительной внешней нормализации. Дополнительная обработка в виде лемматизации, удаления повторов, удаления вложенных единиц и подавления формального шума применяется только на последующих этапах, при работе с автоматически выделенными фразами, псевдометками

и итоговыми предсказаниями. Благодаря этому сопоставление статистических и абстрактных подходов проводится в единых условиях, без изменения исходной формулировки аннотации.

3. Статистико-лингвистическая обработка выделенных фраз

На первом этапе из текста аннотации статистическими методами выделяется набор фраз, потенциально пригодных для использования в качестве ключевых слов. В качестве базового инструмента применяется библиотека статистического извлечения ключевых слов `ruTermExtract`¹, ориентированная на извлечение терминологических словосочетаний из русского текста. Полученный список обычно содержит повторы, вложенные единицы, чрезмерно общие однословные элементы и синтаксически неудачные конструкции. Поэтому после выделения используется статистико-лингвистический фильтр $F(\cdot)$, применяемый ко всему списку кандидатов.

Исходный список выделенных фраз задаётся как

$$C^{raw} = (c_1^{raw}, \dots, c_M^{raw}),$$

а результат фильтрации имеет вид

$$C = F(C^{raw}) = (c_1, \dots, c_{M'}), \quad M' \leq M.$$

Тем самым фильтр преобразует исходный шумный набор кандидатов в более компактный и терминологически устойчивый список, пригодный для последующего использования на этапах построения псевдометок, обучения и итогового формирования ответа.

Фильтр включает несколько последовательно выполняемых операций: нормализацию строк, морфологический анализ, структурную проверку выделенных фраз, статистическое

¹ <https://github.com/igor-shevchenko/rutermextract>

ранжирование, редукцию и удаление вложенных единиц. В совокупности эти шаги предназначены не для точного восстановления эталонной разметки на данном этапе, а для подавления очевидного формального и семантического шума, стабилизации списка кандидатов и выделения его наиболее содержательного ядра.

На первом шаге для каждой фразы строится каноническое представление. Строка переводится в нижний регистр, посторонние символы заменяются пробелами, повторяющиеся пробелы удаляются, после чего выполняются токенизация и лемматизация средствами морфологического анализатора `pymorphy3`². На этой стадии отбрасываются пустые и вырожденные фразы, а также устраняются повторы: сначала по нормализованной поверхностной форме, затем по лемматизированному представлению. Такая двухуровневая схема удаления повторов позволяет учитывать как чисто графические совпадения, так и совпадения, скрытые вариативностью словоформ.

Далее каждая фраза проверяется как приближённая именная группа. Из её начала и конца удаляются служебные или слабоинформативные элементы, после чего оценивается морфологическая структура. В базовом случае сохраняются фразы, у которых последний содержательный токен является существительным, отсутствуют глагольные формы, а морфологический каркас соответствует шаблону

$$(\text{ADJ} \mid \text{NUMR})^* \text{NOUN} ((\text{ADJ} \mid \text{NUMR})^* \text{NOUN})^{0,2}.$$

Такой шаблон допускает не только опорное существительное, но и до двух зависимых именных компонентов после него, каждый из которых может включать определения и завершается существительным. Это позволяет сохранять типичные для научных текстов сочетания вида «метод машинного обучения», «модель обработки данных» или «анализ

² <https://github.com/no-plagiarism/pymorphy3>

временных рядов», в которых после опорного существительного следует одна или две компактные зависимые именные группы. Для фраз, содержащих латиницу, цифры, дефис, подчёркивание или косую черту, используется более мягкая проверка, поскольку такие выражения нередко не укладываются в стандартный морфологический шаблон, но при этом могут представлять собой содержательные термины.

Фразы, прошедшие структурную проверку, получают статистическую оценку, отражающую их терминологическую пригодность. В основе этой оценки лежит предположение о том, что фразы, содержащие более редкие леммы, обладают большей предметной специфичностью, чем сочетания, состоящие из общеупотребительных слов. Для каждой леммы по коллекции аннотаций вычисляется сглаженная обратная документная частота

$$\text{idf}(\ell) = \log \frac{N + 1}{df(\ell) + 1} + 1,$$

где N обозначает число документов корпуса, а $df(\ell)$ — число документов, содержащих лемму ℓ . При вычислении учитываются только содержательные леммы, не относящиеся к служебной лексике.

На уровне фразы используются агрегированные характеристики этих значений, прежде всего средняя и максимальная idf -оценки. Дополнительно учитываются частотные характеристики самой фразы, типичность её морфолого-синтаксического шаблона, длина, степень покрытия леммами исходного текста и лексическая правдоподобность. В результате итоговая оценка кандидата задаётся выражением

$$\begin{aligned} \text{score}(c) = & \left(w_{\text{prior}} f_{\text{prior}}(c) + w_{\text{avg}} f_{\text{idf}}^{\text{avg}}(c) + w_{\text{max}} f_{\text{idf}}^{\text{max}}(c) + w_{\text{pos}} f_{\text{pos}}(c) + w_{\text{len}} f_{\text{len}}(c) \right. \\ & \left. + w_{\text{cov}} f_{\text{cov}}(c) + w_{\text{pres}} f_{\text{pres}}(c) \right) \cdot (0.5 + f_{\text{lex}}(c)), \end{aligned}$$

где f_{prior} отражает частотный приоритет фразы в корпусе, f_{idf}^{avg} и f_{idf}^{max} характеризуют предметную специфичность её лемм, f_{pos} оценивает типичность морфологического шаблона, f_{len} учитывает длину фразы, f_{cov} измеряет степень покрытия кандидата леммами исходного текста, f_{pres} служит регуляризатором для явно присутствующих в тексте единиц, а f_{lex} задаёт общую лексическую уверенность. Весовые коэффициенты w подбирались эвристически на основе корпусной статистики и качественного анализа ошибок.

Для функций, связанных с idf , используется сглаживающее преобразование сигмоидного типа:

$$u(x) = \frac{1}{1 + \exp\left(-\frac{x - \mu_{idf}}{\sigma_{idf}}\right)},$$

где μ_{idf} и σ_{idf} задают положение и масштаб перехода. Это позволяет ограничить влияние экстремально редких лемм и избежать чрезмерного доминирования отдельных выбросов в итоговом ранге кандидата.

Помимо базовой формулы ранжирования, применяются дополнительные штрафы и отсеечения. В частности, понижается оценка фраз, содержащих слишком общие стоп-леммы, чрезмерно частотные выражения или малодостоверные однословные элементы. Для кандидатов, отсутствующих в тексте в явном виде, но обладающих низкой лексической уверенностью, используется мягкий штраф; при ещё более низких значениях соответствующего показателя кандидат может быть отклонён полностью. Такая схема позволяет совместить количественную устойчивость статистического ранжирования с простыми, но полезными ограничениями, направленными на подавление типовых шумовых случаев.

После ранжирования применяется редукция, задача которой состоит не только в удалении шумных фраз, но и в выделении более терминологичного ядра. На этом этапе из фразы могут последовательно удаляться начальные служебные элементы, слабоинформативные определения, чрезмерно общие опорные существительные и крайние токены с низкой специфичностью. Редукция выполняется пошагово и ограничивается небольшим числом итераций, чтобы избежать разрушения исходной структуры содержательных кандидатов. При этом специальные правила защищают опорное существительное и, при необходимости, финальную бигramму типа NOUN NOUN, если она образует устойчивое терминологическое ядро.

Если начальный существительный компонент представлен формой родительного падежа и после редукции оказывается началом итоговой фразы, допускается его нормализация к именительному падежу при сохранении числа. Кроме того, для вариантов, предлагаемых внешними средствами поверхностной нормализации, используется дополнительная проверка на косметические преобразования: модификация принимается только в том случае, если она не искажает исходный набор токенов и действительно приводит к более компактной и содержательной форме кандидата.

На заключительном шаге устраняются вложенные единицы, когда в списке одновременно присутствуют длинное терминологическое сочетание и его более короткая подфраза. В типичном случае более короткая единица удаляется, если она содержится в более длинной и не обладает достаточной самостоятельной терминологической ценностью. Исключения допускаются для некоторых однословных сущностных единиц и для коротких фраз с высокой средней idf-оценкой. Важно, что этот этап выполняется до окончательного слияния результатов: редукция применяется как к исходному набору

кандидатов, так и к его версии после ранжирования, после чего оба варианта объединяются с удалением дубликатов и сохранением исходного порядка. Благодаря этому итоговый список оказывается устойчивее к чрезмерно жёсткому отсечению на одном из промежуточных этапов.

Дополнительно применяется постфильтрация однословных кандидатов. Если некоторая лемма достаточно часто встречается в корпусе как самостоятельный кандидат, но крайне редко совпадает с эталонной разметкой, она рассматривается как слабый однословный элемент и может быть удалена на финальном этапе. Такое решение особенно полезно для подавления чрезмерно общих существительных, регулярно извлекаемых статистическими методами, но практически не выступающих в роли полноценных ключевых слов. При этом используется защитное правило, не допускающее полного опустошения списка кандидатов после постфильтрации.

Работу фильтра можно проиллюстрировать следующим примером. Для одной из аннотаций эталонная разметка имеет вид: *нейрон, нейронная сеть, точечные отображения, особые точки, классификация, фильтрация*. Базовый ruTermExtract возвращает более шумный набор фраз: *нейронная сеть, экспериментальная проверка модели, одномерные точечные отображения, точечные отображения, прямое распространение, пример создания, параметры нейронов, нейронная сеть-классификатор, возможность реализации, результаты, модель, количество*.

Часть кандидатов сохраняется благодаря корректному морфологическому шаблону, а часть отсеивается уже на этапе статистического ранжирования и постфильтрации. После применения фильтра остаётся более компактный и терминологически устойчивый список: *нейронная сеть, экспериментальная проверка модели, одномерные точечные*

отображения, прямое распространение, параметры нейронов, нейронная сеть-классификатор.

В данном случае фильтр удаляет слишком общие и малоспецифичные элементы, такие как *модель, результаты* и *количество*, а также отсеивает часть менее удачных фраз, сохраняя более содержательные терминологические сочетания. При этом его задача состоит не в точном восстановлении эталонной разметки, а в подавлении очевидного шума и стабилизации списка выделенных единиц перед последующими этапами обработки.

Таким образом, статистико-лингвистическая обработка выполняет две взаимосвязанные функции. С одной стороны, она очищает выход статистических методов от дублей, вложенных единиц, слабых однословных кандидатов и формального шума. С другой стороны, она формирует унифицированное и более устойчивое представление выделенных фраз, которое затем используется при построении псевдометок и на последующих этапах обучения.

4. Методы псевдоразметки

В условиях ограниченного объёма размеченных данных при обучении используются не только эталонные пары «аннотация — ключевые слова», но и данные, автоматически сформированные на основе неразмеченной коллекции документов. Документ из этой коллекции задаётся парой $d = (t, a)$, где t обозначает заголовок статьи, а a — её аннотацию. В предлагаемом подходе псевдоразметка используется на нескольких уровнях: для построения целевых последовательностей при предварительном обучении генеративных моделей, для выбора маскируемых фрагментов и для формирования положительных и отрицательных примеров в контрастивном обучении.

Источниками псевдоразметки служат два типа информации. Первый источник является локальным и опирается на сам документ, то есть на его заголовок, аннотацию и выделенные из них фразы. Второй источник является глобальным, поскольку охватывает всю неразмеченную коллекцию и тем самым позволяет включать в обучающие данные семантически близкие единицы, не представленные в текущей аннотации в явном виде. Соответственно, автоматическая разметка может включать как локальную, так и глобальную составляющие.

Исходным этапом построения псевдоразметки служит статистико-лингвистическая фильтрация фраз, выделенных статистическими методами. Отфильтрованный набор используется для формирования псевдометок и выбора маскируемых фрагментов при обучении генеративных моделей, а также для построения положительных и отрицательных примеров в контрастивном обучении и последующем семантическом ранжировании. Таким образом, псевдоразметка в предлагаемой схеме выступает не как отдельный вспомогательный этап, а как общий механизм формирования обучающих данных.

4.1. Локальная псевдоразметка

Локальная псевдоразметка строится по текущему документу $d = (t, a)$. Заголовок и аннотация независимо пропускаются через один и тот же конвейер выделения и статистико-лингвистической фильтрации, после чего из них формируются четыре упорядоченных блока фраз.

Первый блок, $B_1(d)$, содержит фразы из заголовка, явно присутствующие в аннотации. Второй блок, $B_2(d)$, содержит фразы из заголовка, не представленные в аннотации в явном виде, а также допустимые подфразы таких выражений. При сопоставлении с

аннотацией используется лемматизированное представление: фраза считается явно присутствующей в тексте, если её леммы образуют непрерывную подпоследовательность лемм аннотации без пропусков. Для блока $B_1(d)$ внутренний порядок задаётся статистико-лингвистическим фильтром, тогда как для блока $B_2(d)$ применяется дополнительный этап семантического ранжирования на основе векторных представлений, получаемых нейросетевым энкодером семейства E5 [20]; в итоговую последовательность включаются не более десяти таких элементов. Тем самым заголовок даёт два типа опорных единиц: наиболее надёжные, уже выраженные в аннотации, и семантически релевантные, но отсутствующие в ней буквально. Такой способ построения локальной псевдоразметки концептуально близок к подходу, в котором заголовок используется как источник дополнительных фраз для обучения генерации ключевых слов в условиях ограниченной разметки [10].

Далее из аннотации строится ранжированный список выделенных фраз, очищенных статистико-лингвистическим фильтром:

$$R(a) = (r_1, \dots, r_m).$$

Из этого списка исключаются элементы, совпадающие по лемматизированной форме с уже полученными заголовочными метками. Оставшиеся фразы дополнительно ранжируются моделью семейства E5, после чего из верхней части списка формируются ещё два блока:

$$B_3(d) = (r_1, \dots, r_h), \quad B_4(d) = (r_{h+1}, \dots, r_{h+s}).$$

В экспериментах используются значения $h = 5$ и $s = 5$, то есть из верхних десяти фраз аннотации после фильтрации и ранжирования первые пять трактуются как более надёжные документные метки, а следующие пять — как менее очевидные, но всё ещё

содержательные элементы. При этом блок $B_4(d)$ не интерпретируется как строгий аналог ключевых слов, отсутствующих в тексте: входящие в него единицы, как правило, извлекаются из самой аннотации и отражают не буквальное отсутствие, а меньшую очевидность или меньший приоритет соответствующих смысловых компонентов.

Итоговая целевая последовательность локальной псевдоразметки строится как конкатенация четырёх блоков:

$$Y^{loc}(d) = B_1(d) \oplus B_2(d) \oplus B_3(d) \oplus B_4(d),$$

где $\oplus$ обозначает последовательное объединение списков. Тем самым блоки располагаются в порядке убывания предполагаемой надёжности:

$$B_1(d) > B_2(d) > B_3(d) > B_4(d).$$

Перед объединением выполняется удаление повторов: если одна и та же фраза встречается в нескольких блоках, сохраняется только её первое появление в соответствии с указанным порядком приоритета.

Для локальной схемы важна не только целевая последовательность, но и способ построения входа при предварительном обучении. С этой целью вводится операция маскирования. Маскированию подвергаются только элементы множества $B_4(d)$. Для каждой такой фразы в аннотации находятся все вхождения её лемматизированной формы как непрерывной подпоследовательности токенов, после чего каждое вхождение заменяется специальным токеном $\langle mask \rangle$:

$$\tilde{a} = \text{Mask}(a, B_4(d)).$$

Такой выбор принципиален, поскольку маскирование наиболее очевидных и высокорелевантных терминов привело бы к чрезмерному разрушению предметного каркаса аннотации, тогда как маскирование менее очевидных, но содержательных

фрагментов сохраняет основное терминологическое ядро текста и одновременно заставляет модель восстанавливать нетривиальные элементы содержания. Тем самым локальная псевдоразметка одновременно задаёт целевую последовательность и создаёт осмысленный зашумлённый вход для шумоподавляющего предварительного обучения.

Работу локальной схемы можно проиллюстрировать на примере статьи М. С. Дмитриевой и А. В. Забродина «Автоматизация документооборота и бизнес-процессов с помощью ЕСМ-системы Docsvision» [6], аннотация которой посвящена системам управления электронным документооборотом в рабочих процессах делопроизводства. После маскирования элементов блока $B_4(d)$ аннотация принимает вид: «В рамках статьи рассматриваются $\langle mask \rangle$ электронным документооборотом в рабочих процессах делопроизводства. Показаны достоинства и недостатки ЕСМ-систем, которые обеспечивают $\langle mask \rangle$ документами в целях оперативного взаимодействия между структурами внутри организации, $\langle mask \rangle$, а также сокращение временных и трудовых затрат. Рассмотрены $\langle mask \rangle$ электронным документооборотом на примере Docsvision».

Соответствующая локальная целевая последовательность включает, в частности, фразы *docsvision, документооборот, недостатки ЕСМ-систем, электронный документооборот, рабочие процессы делопроизводства, ЕСМ-система Docsvision, автоматизация документооборота, системы Docsvision, бизнес-процессы, система управления, последнее поколение систем управления, улучшение управления, конфиденциальность информации*. Этот пример отражает общий принцип построения локальных меток: заголовок даёт более надёжные опорные элементы, а менее очевидные фразы из аннотации используются как объект маскирования и последующего восстановления.

4.2. Глобальная псевдоразметка

Глобальная псевдоразметка рассматривается как способ дополнения локальных обучающих данных семантически близкими фразами, отсутствующими в текущей аннотации, но потенциально релевантными для генерации ключевых слов, не выраженных в тексте напрямую. В этой схеме для каждого документа сначала формируется объединённое текстовое представление на основе заголовка и аннотации. Затем для документа и для фраз, выделенных из всей неразмеченной коллекции, строятся векторные представления с помощью той же модели семейства E5:

$$e_d = E5(t \oplus a), \quad e_p = E5(p),$$

где p обозначает фразу-кандидат из внешней коллекции.

По этим представлениям выполняется поиск в индексе HNSW [14], построенном по фразам, извлечённым из всей неразмеченной коллекции. Формально близость между документом и фразой определяется косинусным сходством их векторных представлений:

$$s(d, p) = \cos(e_d, e_p).$$

Из найденных элементов отбираются только те фразы, которые не встречаются в аннотации как непрерывные лемматизированные подпоследовательности. Число таких глобальных меток ограничивается значением g . Если обозначить множество отобранных фраз через $G(d)$, то глобальная часть псевдоразметки записывается как

$$G(d) = \text{TopK}_g(\{p: s(d, p) \text{ максимально, } p \notin a\}),$$

где условие $p \notin a$ означает буквальное отсутствие фразы p в тексте аннотации. Полная целевая последовательность в этом случае принимает вид

$$Y^{glob}(d) = Y^{loc}(d) \oplus G(d).$$

Для приведённого выше документа глобальный поиск может возвращать, например, фразы *документооборот, делопроизводство, электронный документооборот, автоматизация документооборота, бизнес-процессы*. Такие элементы семантически связаны с содержанием документа, но не обязательно представлены в нём в той же лексической форме. Вместе с тем у этой схемы есть очевидное ограничение: без дополнительного контроля разнообразия и релевантности глобальное расширение может включать почти дублирующие или семантически избыточные элементы. Поэтому в данной работе глобальная псевдоразметка рассматривается прежде всего как исследуемый способ расширения охвата, а не как заведомо более точный источник обучающих данных по сравнению с локальной схемой.

4.3. Формирование примеров для контрастивного обучения

Псевдоразметка используется не только для построения целевых последовательностей при предварительном обучении генеративных моделей, но и для формирования пространства выделенных фраз, на основе которого строятся положительные и отрицательные примеры в контрастивном обучении. Для каждого документа после статистико-лингвистической обработки формируется очищенный список фраз

$$C(d) = (c_1, \dots, c_m).$$

На размеченном корпусе положительными считаются те фразы из этого списка, которые после нормализации совпадают с эталонными ключевыми словами, явно присутствующими в тексте, и могут быть сопоставлены с непрерывными фрагментами аннотации. Если обозначить множество таких фраз через $C^+(d)$, то множество отрицательных примеров определяется как

$$C^-(d) = C(d) \setminus C^+(d).$$

Среди отрицательных примеров дополнительно выделяются трудные отрицательные элементы, то есть единицы, близкие документу или положительным примерам по форме либо по содержанию, но не являющиеся корректными ответами. Тем самым один и тот же очищенный список кандидатов связывает генеративную и контрастивную составляющие обучения.

Эта схема может быть проиллюстрирована примером аннотации, посвящённой интеллектуальной системе автоматизированного проведения конференций. После выделения и фильтрации формируется набор фраз, включающий, в частности, *Nokia, автоматизированное проведение конференций, взаимодействие участников, интеллектуальная система, интерактивная система проведения конференций, компания Nokia, компонент системы, персональные мобильные устройства, персональные компьютеры, платформа Smart-M3, программа конференции, сотовые телефоны, телефоны*. Если эталонные ключевые слова, явно присутствующие в тексте, представлены, например, словами *конференции* и *онтологии*, то соответствующие выделенные фразы используются как положительные примеры для контрастивной части модели. Остальные очищенные фразы служат отрицательными или трудными отрицательными примерами. При этом эталонное ключевое слово, не представленное в тексте в явном виде, например *интеллектуальное пространство*, используется как целевой объект для генеративной части модели.

Псевдоразметка в предлагаемом подходе используется на трёх уровнях. Во-первых, задаёт целевые последовательности для предварительного обучения базовых генеративных моделей. Во-вторых, использует внешнюю коллекцию для расширения обучающих данных и тем самым делает предварительное обучение более согласованным с

постановкой задачи генерации ключевых слов, не представленных в тексте в явном виде. В-третьих, формирует список выделенных фраз, а также положительные, отрицательные и трудные отрицательные примеры для контрастивных моделей. Благодаря этому псевдоразметка выступает не как узкая процедура автоматического построения целевых строк, а как общий механизм формирования обучающих данных для различных компонентов системы.

5. Модели и схема обучения

Построенная на предыдущем этапе псевдоразметка определяет схему обучения системы: базовая генеративная модель сначала предварительно обучается на неразмеченном корпусе, затем дообучается на размеченных данных, а в конфигурации Generator-Ranker дополняется этапом семантического ранжирования. Очищенные списки выделенных фраз при этом используются для формирования положительных и отрицательных примеров в контрастивном обучении. Общая схема этой конфигурации показана на рис. 1.

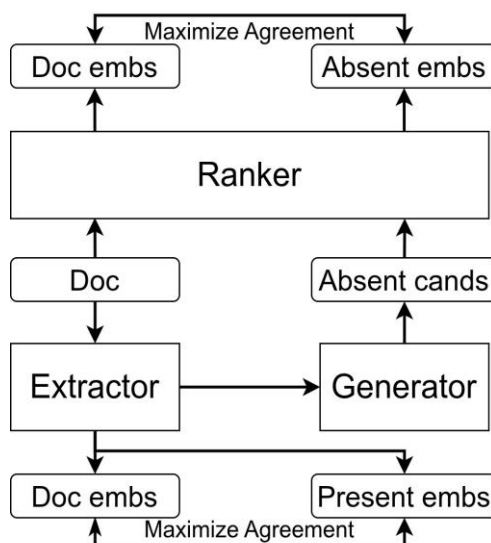


Рис. 1. Общая схема конфигурации Generator-Ranker.

Концептуально предлагаемая архитектура продолжает развитие генеративно-контрастивных схем для генерации ключевых слов, в которых сочетаются последовательностное порождение кандидатов, их выделение из текста и последующее семантическое ранжирование [4]. В данной работе эта идея адаптируется к русскоязычным научным аннотациям и дополняется многоэтапной псевдоразметкой, статистико-лингвистической фильтрацией и отдельной моделью семантического ранжирования для итогового отбора кандидатов.

5.1. Предварительное обучение на псевдоразметке

Пусть $\mathcal{U}$ обозначает неразмеченную коллекцию документов, а $\mathcal{D}$ — размеченный корпус. Предварительное обучение базовой генеративной модели на псевдоразметке задаётся как минимизация последовательностной функции потерь

$$\theta^{(0)} = \operatorname{argmin}_{\theta} \mathcal{L}_{pre}(\theta; \mathcal{U}), \quad \mathcal{L}_{pre}(\theta; \mathcal{U}) = - \sum_{(x, \tilde{Y}) \in \mathcal{U}} \log p_{\theta}(\tilde{Y} | x),$$

где x обозначает входной текст на этапе псевдообучения, а $\tilde{Y}$ — автоматически построенную целевую последовательность.

В локальной схеме на вход модели подаётся замаскированная аннотация $x = \tilde{a}$, а в качестве целевой последовательности используется локальная псевдоразметка $\tilde{Y} = Y^{loc}(d)$. Тем самым предварительное обучение принимает вид шумоподавляющего восстановления: модель получает частично искажённую аннотацию и должна восстановить содержательные фразы, удалённые при маскировании. В глобальной схеме, напротив, на вход подаётся исходная аннотация $x = a$, а в качестве целевого объекта используется расширенная последовательность $\tilde{Y} = Y^{glob}(d)$, включающая как локальную, так и глобальную составляющие. В этом случае предварительное обучение

приближает модель к задаче порождения ключевых слов, не представленных в тексте в явном виде.

После этапа псевдообучения базовая генеративная модель дообучается на размеченном корпусе в стандартной последовательностной постановке:

$$\theta^* = \operatorname{argmin}_{\theta} \mathcal{L}_{sup}(\theta; \mathcal{D}), \quad \mathcal{L}_{sup}(\theta; \mathcal{D}) = - \sum_{(a,Y) \in \mathcal{D}} \log p_{\theta}(Y | a),$$

где Y обозначает эталонный набор ключевых слов для аннотации a . Полученные таким образом модели рассматриваются либо как самостоятельные базовые методы, либо как инициализация для более сложной архитектуры.

5.2. Генеративно-контрастивная модель

Следующим компонентом является генеративно-контрастивная модель, построенная на основе архитектуры энкодер-декодер. В этой конфигурации общий энкодер используется одновременно для двух задач: выделения ключевых слов, явно присутствующих в тексте, и генерации ключевых слов, не представленных в нём в явном виде. Генеративно-контрастивная модель инициализируется весами базовой генеративной модели, предварительно обученной на локальной псевдоразметке. Тем самым контрастивный модуль строится не поверх случайной инициализации, а на основе уже адаптированного к задаче генеративного представления. При этом параметры энкодера и декодера наследуются полностью, тогда как дополнительные проекции для контрастивной части обучаются с нуля. Архитектура этого модуля показана на рис. 2.

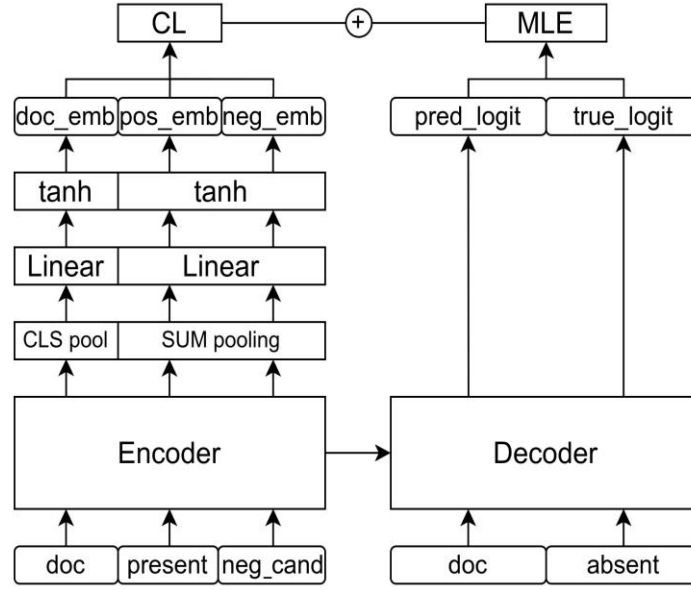


Рис. 2. Архитектура генеративно-контрастивной модели (Extractor-Generator).

Пусть после токенизации входная аннотация имеет вид

$$x = (x_1, \dots, x_n),$$

а после прохождения через энкодер получается последовательность скрытых состояний

$$H = (h_1, \dots, h_n).$$

Для той же аннотации строится очищенный список выделенных фраз

$$C(a) = (c_1, \dots, c_m).$$

Те из этих фраз, которые после нормализации могут быть сопоставлены с непрерывными фрагментами текста, используются в контрастивной части модели. Если такая фраза соответствует промежутку токенов (s, e) , то этот фрагмент рассматривается как её текстовое вхождение в аннотации.

Для представления документа используется нормированный вектор, построенный по скрытому состоянию токена CLS:

$$z_a = \text{norm}(W_d h_{\text{CLS}} + b_d).$$

Для фрагмента (s, e) сначала вычисляется усреднённое представление его токенов

$$\bar{h}_{s:e} = \frac{1}{e - s + 1} \sum_{i=s}^e h_i,$$

после чего применяется линейная проекция и нормирование:

$$z_{s:e} = \text{norm}(W_p \bar{h}_{s:e} + b_p).$$

Тем самым и документ, и кандидатная фраза приводятся к одному векторному пространству, в котором их близость измеряется косинусной мерой

$$s(a, s:e) = \cos(z_a, z_{s:e}).$$

На размеченном корпусе положительными примерами считаются те фразы, которые совпадают с эталонными ключевыми словами, явно присутствующими в тексте, и могут быть сопоставлены с непрерывными фрагментами аннотации. Остальные допустимые фразы образуют множество отрицательных примеров. Поскольку такие отрицательные примеры представляют собой грамматически корректные словосочетания, извлечённые из того же документа, они существенно сложнее случайных строк и приближаются по характеру к трудным отрицательным примерам, используемым в контрастивных постановках. Контрастивная функция потерь имеет вид

$$\mathcal{L}_{ctr}(a) = -\log \frac{\sum_{(s,e) \in \mathcal{K}^+(a)} \exp\left(\frac{s(a, s:e)}{\tau}\right)}{\sum_{(s,e) \in \mathcal{K}^+(a) \cup \mathcal{K}^-(a)} \exp\left(\frac{s(a, s:e)}{\tau}\right)}$$

где $\mathcal{K}^+(a)$ и $\mathcal{K}^-(a)$ обозначают множества положительных и отрицательных фрагментов соответственно, а $\tau > 0$ — температурный параметр.

Параллельно с этим декодер обучается на генерацию ключевых слов, не представленных в тексте в явном виде. Пусть $y^A = (y_1^A, \dots, y_T^A)$ обозначает упорядоченную последовательность эталонных ключевых слов этой группы. Тогда генеративная часть оптимизируется по стандартной функции максимального правдоподобия:

$$\mathcal{L}_{gen}(a, y^A) = - \sum_{t=1}^T \log p_{\theta}(y_t^A \mid y_{<t}^A, a).$$

Совместная функция потерь для генеративно-контрастивной модели записывается как

$$\mathcal{L} = \gamma \mathcal{L}_{ctr} + (1 - \gamma) \mathcal{L}_{gen},$$

где $\gamma \in [0,1]$ задаёт баланс между контрастивной и генеративной составляющими.

Важно, что контрастивная часть модели не обучается на локальной псевдоразметке. Автоматически построенные метки не гарантируют достаточно надёжных границ фраз в тексте, тогда как в контрастивной постановке именно корректность таких границ имеет принципиальное значение. Поэтому контрастивное обучение проводится только на размеченном корпусе, где соответствие между фразами и их текстовыми вхождениями задано существенно надёжнее.

5.3. Модель семантического ранжирования

Для конфигурации Generator-Ranker используется отдельная модель семантического ранжирования, реализованная в виде биэнкодера на базе E5. Модель получает на вход объединённый набор фраз, сформированный генеративно-контрастивной моделью, и выполняет итоговое семантическое ранжирование кандидатов относительно содержания документа.

Архитектура модели семантического ранжирования в своей основе повторяет энкодерную часть контрастивного компонента генеративно-контрастивной модели и отличается прежде всего использованием E5 в качестве базовой модели.

На вход модели семантического ранжирования поступают фразы из двух источников.

Первый источник — фразы, порождённые декодером генеративно-контрастивной модели.

Второй источник — фразы, выделенные контрастивным компонентом генеративно-

контрастивной модели. Если обозначить соответствующие множества через $U_{gen}(a)$ и $U_{ext}(a)$, то итоговый список задаётся как

$$U(a) = U_{gen}(a) \cup U_{ext}(a).$$

Модель семантического ранжирования кодирует документ и фразу независимо:

$$q(a) = E_d(a), \quad r(k) = E_k(k),$$

где $E_d(\cdot)$ и $E_k(\cdot)$ обозначают энкодерные преобразования аннотации и фразы соответственно. После нормировки их близость оценивается той же косинусной мерой, что и выше:

$$s(a, k) = \cos(q(a), r(k)).$$

На размеченном корпусе положительными считаются те элементы множества $U(a)$, которые совпадают с эталонными ключевыми словами после нормализации. Остальные фразы используются как отрицательные примеры. Тем самым модель обучается различать действительно релевантные ключевые слова и семантически близкие, но нерелевантные варианты.

На этапе применения системы генеративно-контрастивная модель формирует два типа выходов. Декодер с помощью лучевого поиска порождает ключевые слова, не представленные в тексте в явном виде, а контрастивная часть выделяет наиболее релевантные фразы, явно присутствующие в аннотации. После этого модель семантического ранжирования упорядочивает объединённый список $U(a)$ по величине $s(a, k)$, а к отобранным верхним элементам повторно применяется статистико-лингвистический фильтр, описанный в разделе 3. Тем самым итоговая схема объединяет три источника сигнала: псевдоразметку как механизм предварительного обучения,

генеративно-контрастивную модель как средство совместного выделения и порождения ключевых слов и модель семантического ранжирования как завершающий этап отбора.

6. Эксперименты

6.1. Данные

Экспериментальная оценка проводилась на размеченном корпусе *Math&CS*³, содержащем аннотации научных статей по математике и компьютерным наукам. Корпус включает 8348 документов, из которых 5844 используются для обучения, а 2504 — для тестирования. В среднем на один документ приходится 4.34 ± 1.50 эталонных ключевых слова. Существенной особенностью корпуса является высокая доля ключевых слов, не представленных в тексте аннотации в явном виде: 53.66% эталонных единиц относятся к этой категории. Тем самым задача генерации оказывается принципиально не сводимой к простому извлечению слов и словосочетаний из текста.

Основные характеристики корпуса приведены в табл. 1. Число токенов рассчитывалось после токенизации с помощью библиотеки NLTK⁴.

Таблица 1. Характеристики корпуса Math&CS

Характеристика	Значение
Размер обучающей выборки	5844
Размер тестовой выборки	2504
Среднее число предложений	3.73 ± 2.75
Среднее число токенов	74.16 ± 61.65
Среднее число ключевых слов	4.34 ± 1.50
Доля ключевых слов, отсутствующих в тексте аннотации	53.66%

3 https://huggingface.co/datasets/aglazkova/keyphrase_extraction_russian

4 <https://www.nltk.org/>

Для построения псевдоразметки и предварительного обучения моделей использовался неразмеченный корпус ruSciBench⁵, содержащий более 190 тысяч аннотаций научных статей. Этот корпус применялся только для построения локальной и глобальной псевдоразметки и не использовался при вычислении итоговых метрик. Его основные характеристики приведены в табл. 2, где число ключевых слов рассчитывалось по результатам их автоматического извлечения с помощью библиотеки ruTermExtract.

Таблица 2. Характеристики корпуса ruSciBench

Характеристика	Значение
Размер корпуса	194071
Среднее число предложений	4.51 ± 0.05
Среднее число токенов	91.54 ± 1.20
Среднее число ключевых слов	4.51 ± 0.02

6.2. Настройки обучения и генерации

В качестве базовых генеративных моделей использовались архитектуры типа энкодер-декодер mT5-base⁶ (580M параметров), mBART-large⁷ (610M) и ru-mbart-large-summ⁸ (380M). В дальнейшем в качестве основной базовой модели использовалась ru-mbart-large-summ (далее — ru-mbart-summ). По своей архитектуре она относится к семейству mBART и опирается на русско-английскую конфигурацию, полученную из mBART-large путём сокращения числа поддерживаемых языков и уменьшения словаря SentencePiece с 250 тыс. до 25 тыс. наиболее частотных русских и английских токенов. Благодаря этому

5 https://huggingface.co/datasets/mlsa-iai-msu-lab/ru_sci_bench

6 <https://huggingface.co/google/mt5-base>

7 <https://huggingface.co/facebook/mBART-large-50>

8 <https://huggingface.co/d0rj/ru-mBART-large-summ>

ru-mbart-summ сохраняет преимущества архитектуры mBART, оставаясь при этом более компактной моделью, ориентированной на обработку русскоязычных текстов. Существенно и то, что данная модель уже была дообучена на корпусах суммаризации, то есть адаптирована к задаче условной генерации коротких содержательных последовательностей, близкой к генерации ключевых слов.

Модели обучались в течение 10 эпох; максимальная длина входной последовательности составляла 256 токенов. Для оптимизации использовался алгоритм Adam с параметрами $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 4 \cdot 10^{-8}$, при скорости обучения $lr = 4 \cdot 10^{-5}$. Для дополнительных параметров, связанных с контрастивной частью модели, использовался AdamW с коэффициентом регуляризации $weight_decay = 0.01$.

Все эти настройки соответствуют параметрам, зафиксированным в текущей экспериментальной конфигурации.

Генерация ключевых слов выполнялась с помощью лучевого поиска. Использовались параметры декодирования `num_beams = 100`, `repetition_penalty = 1.4` и `no_repeat_ngram_size = 2`. В конфигурациях, где после генерации применялось дополнительное семантическое ранжирование, генеративная модель формировала расширенный набор фраз (`num_return_sequences = 50`), который затем передавался на вход модели семантического отбора. При тестировании в конфигурации Generator итоговый список формировался из двух источников: энкодер отбирал 5 фраз, явно присутствующих в тексте, а декодер порождал 5 ключевых слов, не представленных в нём в явном виде. Для остальных моделей и статистических методов в итоговую оценку включались первые 10 предсказанных кандидатов.

Контрастивное обучение выполнялось с температурным параметром $\tau = 0.1$ и коэффициентом $\gamma = 0.3$, определяющим баланс между контрастивной и генеративной составляющими функции потерь. В качестве модели семантического ранжирования использовалась компактная двуязычная версия модели семейства E5 —e5-large-en-ru⁹ (366M). При вычислении метрики BERTScore применялась модель bert-base-multilingual-cased¹⁰.

6.3. Сравниваемые конфигурации

В экспериментах рассматривались несколько групп методов. К первой группе относятся базовые экстрактивные алгоритмы YAKE!¹¹ и ruTermExtract, извлекающие ключевые слова непосредственно из текста аннотации без использования обучаемых нейросетевых моделей. Для оценки вклада статистико-лингвистической обработки эти методы анализировались как в исходной конфигурации, так и после применения предложенного фильтра.

Ко второй группе относятся базовые генеративные модели mT5-base (580M), mBART-large (610M) и ru-mbart-summ (380M), обученные только на размеченном корпусе без предварительного обучения на псевдоразметке. Они используются как отправная точка для сравнения с конфигурациями, в которых привлекаются дополнительные обучающие данные, сформированные на основе неразмеченных данных.

9 <https://huggingface.co/d0rj/e5-large-en-ru>

10 <https://huggingface.co/google-bert/bert-base-multilingual-cased>

11 <https://github.com/boudinfl/pke>

Выбор ru-mbart-summ в качестве основной модели для дальнейших модификаций обусловлен тем, что она, сохраняя архитектурную основу mBART, остаётся более компактной и пригодной для поэтапного анализа вклада предлагаемых методов. Модель mBART-large при этом используется в работе как более сильная базовая модель, с которой сопоставляются результаты компактной модели.

Третью группу составляют конфигурации, основанные на предложенных в работе методах псевдоразметки. В этих экспериментах модель ru-mbart-summ предварительно обучается на автоматически построенных метках, после чего дообучается на размеченном корпусе. Отдельно анализируются локальная схема с маскированием и глобальная схема, использующая семантически близкие фразы из внешней коллекции.

Наконец, отдельно рассматриваются конфигурации Generator и Generator-Ranker, в которых предварительно обученная генеративная модель дополняется контрастивным модулем выделения фраз и, при необходимости, отдельным этапом семантического ранжирования на базе E5. Это позволяет разделить вклад локальной псевдоразметки, генеративно-контрастивного компонента и завершающего смыслового отбора кандидатов. В таблицах с результатами для краткости используются условные англоязычные обозначения конфигураций. Filter обозначает применение предложенного статистико-лингвистического фильтра к выходу экстрактивного метода. Local pseudo-labeling соответствует предварительному обучению модели ru-mbart-summ на локальной псевдоразметке с маскированием. Global pseudo-labeling обозначает предварительное обучение той же модели на расширенной псевдоразметке, включающей глобальную составляющую. Generator соответствует генеративно-контрастивной конфигурации, а

Generator-Ranker — той же конфигурации, дополненной этапом семантического ранжирования.

6.4. Метрики оценки

Качество системы оценивалось по первым десяти предсказанным ключевым словам. Если модель возвращала более десяти элементов, в расчёт принимались только первые десять; если меньше, использовался весь полученный список. Перед вычислением метрик предсказанные и эталонные ключевые слова приводились к сопоставимому виду: приведение к нижнему регистру, лемматизация и удаление внешних знаков пунктуации. Точное совпадение фиксировалось только в том случае, если лемматизированные последовательности токенов полностью совпадали.

В качестве основной метрики точного совпадения использовалась $F1@10$, вычисляемая для каждого документа по множеству первых десяти предсказанных ключевых слов и множеству эталонных единиц как гармоническое среднее точности и полноты. Далее рассматривались две схемы агрегирования этой метрики по тестовой выборке. $Micro-F1@10$ вычислялась после объединения предсказаний и эталонных множеств всех документов тестовой выборки. $Macro-F1@10$, напротив, определялась как среднее значение $F1@10$ по отдельным документам и потому чувствительнее к разбросу качества между текстами разной сложности.

Дополнительно рассчитывались отдельные микроусреднённые значения $F1$ для двух групп эталонных ключевых слов: присутствующих в тексте (далее — $F1$ Pres.) и не представленных в нём в явном виде (далее — $F1$ Abs.). Ключевое слово относилось к первой группе, если его лемматизированная последовательность образовывала непрерывную подпоследовательность лемм аннотации без пропусков; во всех остальных

случаях оно относилось ко второй группе. Это позволяло отдельно анализировать способность модели извлекать явно выраженные единицы и порождать более абстрактные формулировки. Macro-F1@10 использовалась как дополнительная диагностическая характеристика; поскольку она не изменяла порядок лидирующих конфигураций и не влияла на интерпретацию результатов, в основных таблицах для компактности приводятся микроусреднённые значения.

Для оценки частичного лексического совпадения использовалась метрика ROUGE-1@10 [13]. В этом случае списки предсказанных и эталонных ключевых слов для каждого документа конкатенировались в две строки, после чего между ними вычислялась F1-мера перекрытия униграмм. Такая постановка позволяет учитывать не только полные совпадения фраз, но и частичное совпадение содержательных слов внутри списков. Семантическое качество предсказаний оценивалось с помощью BERTScore@10 [21]. Как и в случае ROUGE, списки предсказанных и эталонных ключевых слов предварительно объединялись в строки, после чего между этими строками вычислялась семантическая F1-мера на основе косинусного сходства контекстных эмбеддингов модели bert-base-multilingual-cased. Эта метрика особенно важна для рассматриваемой задачи, поскольку позволяет учитывать случаи, когда модель порождает семантически близкую, но не совпадающую дословно формулировку. По этой причине именно BERTScore@10 рассматривалась как основная метрика при сравнении моделей, тогда как F1 использовалась прежде всего для анализа точных совпадений и отдельной оценки двух типов ключевых слов. В качестве дополнительной диагностической характеристики измерялось среднее число предсказанных ключевых слов, позволяющее оценить

склонность модели к избыточной генерации и качество последующей фильтрации и семантического отбора.

Статистическая значимость различий оценивалась методом парного бутстрэпа, широко применяемым при сравнении систем автоматической обработки текста [7, 11]. При интерпретации результатов учитывались общие рекомендации по проверке статистической значимости в задачах автоматической обработки естественного языка [1, 7]. Для каждой метрики по документам тестовой выборки проводилось 2000 итераций бутстрэпа с выборкой документов с возвращением. При попарном сравнении моделей А и В на каждой итерации бутстрэпа вычислялась разность $\Delta = M(A) - M(B)$, где M — значение соответствующей метрики. По распределению разностей строился 95%-ный доверительный интервал методом ВСа (bias-corrected and accelerated), то есть с поправкой на смещение и ускорение, на основе скорректированных квантилей, соответствующих номинальному уровню 95%. Различие считалось статистически значимым, если доверительный интервал для Δ не содержал нуль, то есть если оба его конца имели один и тот же знак. Если интервал включал нуль, различие интерпретировалось как статистически незначимое. Основной вывод о статистической значимости делался по метрике BERTScore@10, учитывающей семантическую близость предсказаний, тогда как интервалы для Micro-F1@10 и F1 Abs. использовались как дополнительная диагностическая информация и приведены отдельно вместе с интервалами для BERTScore@10.

7.1. Сравнительный анализ методов

Полученные результаты показывают, что статистико-лингвистическая фильтрация и предварительное обучение на псевдоразметке заметно влияют на качество генерации ключевых слов. В целом предложенные модификации улучшают результаты по сравнению с экстрактивными методами и моделью ru-mbart-summ, обученной только на размеченном корпусе; вместе с тем mBART-large демонстрирует наиболее высокие значения по совокупности основных метрик.

Модель	F1@10	F1 Pres.	F1 Abs.	R1@10	BS@10	Среднее число предсказанных ключевых слов
Экстрактивные методы						

YAKE!	10.52	11.34	-	30.06	74.22	11.85
YAKE! + Filter	12.02	14.49	-	32.95	75.29	7.86
ruTermExtract	11.05	10.90	-	30.65	76.14	15.10
ruTermExtract + Filter	<i>13.12</i>	<i>15.17</i>	-	<i>34.14</i>	<i>77.09</i>	9.77
Генеративные модели						
mT5-base	11.94	15.09	0.67	31.71	77.46	4.51
mBART-large	19.65	26.24	1.06	40.76	80.09	3.22
ru-mbart-summ	13.01	17.49	0.89	33.58	78.30	2.96
Предлагаемые подходы						
ru-mbart-summ + Local pseudo-labeling	<i>18.16</i>	<i>24.81</i>	0.69	39.13	79.61	3.02
ru-mbart-summ + Global pseudo-labeling	16.79	23.10	0.64	38.47	79.19	2.90
ru-mbart-summ + Generator	14.71	16.82	0.92	34.83	77.64	8.00
ru-mbart-summ + Generator-Ranker	15.29	17.00	1.47	36.45	78.07	7.89

Среди экстрактивных методов наилучшие результаты демонстрирует конфигурация ruTermExtract + Filter: $F1@10 = 13.12$, $R1@10 = 34.14$ и $BS@10 = 77.09$. Прирост относительно исходного ruTermExtract подтверждает, что статистико-лингвистическая обработка повышает устойчивость списка выделенных кандидатов. Сходный эффект наблюдается и для YAKE!: после фильтрации показатели возрастают с 10.52 до 12.02 по $F1@10$ и с 74.22 до 75.29 по $BS@10$. Следовательно, уже на уровне статистических методов фильтр не только уменьшает шум, но и повышает качество итоговых предсказаний.

По сравнению с экстрактивными методами базовые генеративные модели демонстрируют более высокое семантическое качество. Даже без предварительного обучения на псевдоразметке модель ru-mbart-summ превосходит лучший экстрактивный базовый уровень ruTermExtract + Filter по метрике $BS@10$: 78.30 против 77.09. Согласно

результатам парного бутстрэпа, различие статистически значимо, а 95%-ный доверительный интервал для разности составляет [1.01; 1.38]. Следовательно, базовая генеративная постановка обеспечивает более точное семантическое соответствие предсказаний содержанию документа, чем статистическое извлечение фраз. В то же время модель mT5-base показывает более скромные результаты и заметно уступает даже базовой конфигурации ru-mbart-summ, что указывает на лучшую пригодность архитектуры mBART для рассматриваемой задачи.

Наиболее сильной среди сопоставимых конфигураций выступает модель ru-mbart-summ + Local pseudo-labeling, предварительно обученная на локальной псевдоразметке с маскированием. Значения основных метрик для данной конфигурации составляют $F1@10 = 18.16$, $R1@10 = 39.13$ и $BS@10 = 79.61$, что заметно выше результатов базовой модели ru-mbart-summ. Улучшение по $BS@10$ достигает 1.31, а результаты парного бутстрэпа показывают статистически значимое превосходство при 95%-ном доверительном интервале [1.13; 1.48]. Таким образом, локальная псевдоразметка с маскированием обеспечивает наилучшие результаты при использовании в качестве дополнительных обучающих данных для генеративной модели.

Отдельного внимания заслуживает модель mBART-large, демонстрирующая среди генеративных базовых конфигураций наилучшие численные результаты по всем основным агрегированным метрикам. Вместе с тем предварительное обучение на локальной псевдоразметке существенно сокращает разрыв между компактной моделью ru-mbart-summ и более крупной mBART-large: по метрике $BS@10$ разрыв уменьшается с 1.79 до 0.48 пункта. Локальная псевдоразметка, таким образом, не устраняет отставание

полностью, но заметно приближает компактную модель к более сильному генеративному базовому уровню.

7.2. Влияние статистико-лингвистической фильтрации и псевдоразметки

95%-ные доверительные интервалы по основным метрикам приведены в табл. 4. Интервальные оценки позволяют судить об устойчивости наблюдаемых различий по точному совпадению, генерации отсутствующих ключевых слов и семантическому качеству.

Таблица 4. 95%-ные доверительные интервалы для основных метрик

Модель	95%-ный ДИ F1@10	95%-ный ДИ F1 Abs.	95%-ный ДИ BS@10
Экстрактивные методы			
YAKE!	[10.03, 11.00]	-	[74.05, 74.38]
YAKE! + Filter	[11.47, 12.60]	-	[75.11, 75.46]
ruTermExtract	[10.54, 11.56]	-	[75.98, 76.31]
ruTermExtract + Filter	[12.56, 13.69]	-	[76.91, 77.28]
Генеративные модели			
mT5-base	[11.24, 12.41]	[0.51, 0.88]	[77.25, 77.66]
mBART-large	[18.83, 20.45]	[0.83, 1.33]	[79.88, 80.33]
ru-mbart-summ	[12.25, 13.71]	[0.69, 1.14]	[78.10, 78.51]
Предлагаемые подходы			
ru-mbart-summ + Local pseudo-labeling	[17.39, 18.98]	[0.51, 0.92]	[79.39, 79.84]
ru-mbart-summ + Global pseudo-labeling	[16.04, 17.60]	[0.46, 0.87]	[78.97, 79.42]
ru-mbart-summ + Generator	[14.12, 15.31]	[0.75, 1.11]	[77.48, 77.82]
ru-mbart-summ + Generator-Ranker	[14.69, 15.93]	[1.26, 1.69]	[77.90, 78.24]

Результаты попарных сравнений приведены в табл. 5. В данной таблице «Средняя разность, Δ » обозначает среднюю разность значений метрик, а столбец «95%-ный ДИ Δ »

— доверительный интервал с поправкой на смещение и ускорение. Такой способ представления позволяет отделить вклад отдельных компонентов системы от общих различий между классами моделей.

Таблица 5. Попарные сравнения моделей по основным метрикам

Метрика	Модель А	Модель В	Средняя разность, Δ	95%-ный ДИ Δ	Вывод
Влияние фильтрации					
BS@10	YAKE! + Filter	YAKE!	+1.07	[1.00, 1.14]	A > B
F1@10	YAKE! + Filter	YAKE!	+1.49	[1.28, 1.81]	A > B
BS@10	ruTermExtract + Filter	ruTermExtract	+0.96	[0.89, 1.02]	A > B
F1@10	ruTermExtract + Filter	ruTermExtract	+2.09	[1.81, 2.32]	A > B
Генеративные и экстрактивные методы					
BS@10	ru-mbart-summ	ruTermExtract + Filter	+1.20	[1.01, 1.38]	A > B
BS@10	ru-mbart-summ	YAKE! + Filter	+3.01	[2.83, 3.19]	A > B
Влияние локальной псевдоразметки					
BS@10	ru-mbart-summ + Local pseudo-labeling	ru-mbart-summ	+1.32	[1.13, 1.48]	A > B
F1 Abs.	ru-mbart-summ + Local pseudo-labeling	ru-mbart-summ	-0.20	[-0.46, 0.02]	различия незначимы
Влияние глобальной псевдоразметки					
BS@10	ru-mbart-summ + Global pseudo-labeling	ru-mbart-summ + Local pseudo-labeling	-0.43	[-0.57, -0.27]	B > A
F1 Abs.	ru-mbart-summ + Global pseudo-labeling	ru-mbart-summ + Local pseudo-labeling	-0.05	[-0.21, 0.11]	различия незначимы
Влияние конфигураций Generator					
BS@10	ru-mbart-summ + Generator	ru-mbart-summ + Local pseudo-labeling	-1.97	[-2.12, -1.78]	B > A

F1 Abs.	ru-mbart-summ + Generator	ru-mbart-summ + Local pseudo-labeling	+0.23	[-0.01, 0.46]	различия незначимы
Влияние конфигураций Generator-Ranker					
BS@10	ru-mbart-summ + Generator-Ranker	ru-mbart-summ + Generator	+0.42	[0.36, 0.50]	A > B
F1 Abs.	ru-mbart-summ + Generator-Ranker	ru-mbart-summ + Generator	+0.55	[0.35, 0.76]	A > B

Прежде всего, статистико-лингвистическая фильтрация оказывает устойчиво положительное влияние на экстрактивные методы. Для YAKE! применение фильтра даёт прирост +1.07 по BS@10 и +1.49 по F1@10; оба различия статистически значимы. Для ruTermExtract эффект выражен ещё сильнее: прирост составляет +0.96 по BS@10 и +2.09 по F1@10, также при статистически значимом превосходстве отфильтрованной версии. Полученные результаты подтверждают, что предложенный фильтр выполняет не только косметическую, но и функциональную роль, подавляя шум и улучшая качество терминологически значимых кандидатов.

На уровне генеративных моделей наиболее выраженный эффект даёт локальная псевдоразметка. Конфигурация ru-mbart-summ + Local pseudo-labeling статистически значимо превосходит базовую модель ru-mbart-summ по метрике BS@10. По метрике F1 Abs. наблюдается снижение с 0.89 до 0.69, однако доверительный интервал для разности [-0.46, 0.02] включает нуль, поэтому такое различие нельзя считать статистически значимым. Иными словами, локальная схема с маскированием надёжно улучшает общее семантическое качество генерации, но не даёт статистически подтверждённого выигрыша именно в порождении ключевых слов, отсутствующих в тексте в явном виде.

Глобальная псевдоразметка в текущей реализации не даёт преимущества по сравнению с локальной схемой. Конфигурация ru-mbart-summ + Global pseudo-labeling уступает модели

ru-mbart-summ + Local pseudo-labeling по BS@10 на 0.42, причём снижение статистически значимо; доверительный интервал для разности составляет [-0.57, -0.27]. Различие по метрике F1 Abs. при этом остаётся статистически незначимым. Следовательно, расширение обучающих данных за счёт внешней коллекции в данной постановке не приводит к повышению качества и, вероятно, сопровождается включением избыточных либо недостаточно точных глобальных меток.

7.3. Влияние конфигураций Generator и Generator-Ranker

Дополнительный интерес представляет конфигурация Generator-Ranker, сочетающая генеративно-контрастивную модель и отдельный этап семантического ранжирования. Результаты для данной конфигурации оказываются неоднозначными, но содержательно показательными.

Конфигурация ru-mbart-summ + Generator по сравнению с ru-mbart-summ + Local pseudo-labeling ухудшает общий результат по BS@10: снижение составляет 1.97 и является статистически значимым. По метрике F1 Abs., напротив, наблюдается прирост с 0.69 до 0.92, однако доверительный интервал [-0.01, 0.46] включает нуль, поэтому такой рост нельзя считать статистически подтверждённым. Подобная картина показывает, что подключение генеративно-контрастивной схемы делает модель более склонной к расширению множества кандидатов, но без завершающего отбора сопровождается снижением общего семантического качества.

Именно по этой причине ключевым компонентом становится этап семантического ранжирования. Подключение модели семантического ранжирования к конфигурации ru-mbart-summ + Generator даёт статистически значимый прирост как по BS@10 (+0.42, доверительный интервал [0.36, 0.50]), так и по F1 Abs. (+0.55, доверительный интервал

[0.35, 0.76]). Такой результат показывает, что этап семантического ранжирования действительно повышает качество отбора кандидатов и особенно полезен для компоненты ключевых слов, отсутствующих в тексте в явном виде.

Вместе с тем итоговая конфигурация ru-mbart-summ + Generator-Ranker не превосходит ru-mbart-summ + Local pseudo-labeling по общей метрике BS@10: значения составляют 78.07 и 79.61 соответственно. Следовательно, данная конфигурация не является универсально лучшей по всем метрикам, однако решает иную задачу, поскольку заметно усиливает компоненту ключевых слов, отсутствующих в тексте в явном виде. По метрике F1 Abs. модель ru-mbart-summ + Generator-Ranker демонстрирует лучший результат среди всех конфигураций серии ru-mbart-summ, достигая 1.47 против 0.69 у ru-mbart-summ + Local pseudo-labeling и 0.64 у ru-mbart-summ + Global pseudo-labeling. Таким образом, локальная псевдоразметка с маскированием остаётся лучшей стратегией для повышения общего семантического качества, тогда как генеративно-контрастивная модель в конфигурации Generator-Ranker наиболее полезна для расширения множества ключевых слов, отсутствующих в тексте в явном виде.

Различие между стратегиями дополнительно подтверждается показателем среднего числа предсказанных ключевых слов. Модели ru-mbart-summ, ru-mbart-summ + Local pseudo-labeling и ru-mbart-summ + Global pseudo-labeling возвращают в среднем около трёх ключевых слов на документ, тогда как ru-mbart-summ + Generator и ru-mbart-summ + Generator-Ranker возвращают около восьми. Следовательно, конфигурация Generator-Ranker повышает полноту и охват ценой менее строгой селективности. Локально предобученная базовая модель, напротив, сохраняет большую компактность и более высокую среднюю точность.

7.4. Анализ ошибок

Качественный анализ предсказаний показывает, что различные группы методов совершают разные типы ошибок. Для иллюстрации рассмотрим аннотацию статьи К. Ю. Колыбанова и соавторов [12]:

«Описан процесс создания многоуровневой информационно-аналитической системы, использующей идеи многомерного анализа данных и направленной на обеспечение управлением материально-техническим оснащением образовательных учреждений.»

Предсказания различных моделей для данной аннотации приведены в табл. 6.

Таблица 6. Примеры предсказаний различных моделей для одной аннотации

Модель	Ключевые слова
Эталон	визуализация данных; информационная поддержка принятия управленческих решений; многомерная модель данных; многомерные кубы; многомерные модели; гиперкубы; многоуровневая иерархическая структура; olap-технологии; первичный бухгалтерский учёт; разделение данных на показатели, переменные и измерения; признаки; реляционные модели; система мониторинга материально-технического оснащения; системы подготовки принятия управленческих решений; технологии хранилища данных
YAKE!	многоуровневый информационно-аналитический система; процесс создание; обеспечение управление; материально технический оснащение; образовательный учреждение; идея; многомерный анализ данные
YAKE! + Filter	многоуровневый информационно-аналитический система; обеспечение управление; материально технический оснащение; образовательный учреждение; многомерный анализ данные; технический оснащение
ruTermExtract	многоуровневая информационно-аналитическая система; многомерный анализ данных; процесс создания; образовательные учреждения; материально-техническое оснащение; использующие идеи; управление; обеспечение
ruTermExtract + Filter	многоуровневая информационно-аналитическая система; многомерный анализ данных; образовательные учреждения; материально-техническое оснащение
mT5-base	многомерный анализ данных; информационно-аналитическая

Модель	Ключевые слова
	система; управление материально-техническим оснащением
mBART-large	информационно-аналитическая система; материально-техническое оснащение образовательных учреждений; многомерный анализ данных
ru-mbart-summ	многоуровневая информационная система; многомерный анализ данных; управление материально-техническими учреждениями
ru-mbart-summ + Local pseudo-labeling	многоуровневая информационная система; многомерный анализ данных; материально-техническое оснащение
ru-mbart-summ + Global pseudo-labeling	многоуровневая информационно-аналитическая система; материально-техническое оснащение; образовательные учреждения
ru-mbart-summ + Generator	многомерный анализ данных; многоуровневая информационно-аналитическая система; образовательные учреждения; материально-техническое оснащение; поддержка принятия решений; методы многокритериального программирования
ru-mbart-summ + Generator-Ranker	многомерный анализ данных; многоуровневая информационно-аналитическая система; образовательные учреждения; материально-техническое оснащение; информационные технологии; информационная система; измерительный комплекс

Как видно из табл. 6, экстрактивные методы преимущественно воспроизводят фразы, непосредственно выраженные в тексте аннотации. При этом исходные версии YAKE! и ruTermExtract возвращают заметное количество шумовых или синтаксически неудачных единиц, таких как процесс создание, идея, использующие идеи, обеспечение. После применения статистико-лингвистического фильтра соответствующие списки становятся компактнее и чище, однако по-прежнему остаются в пределах поверхностно выраженной лексики документа.

Генеративные базовые модели ведут себя иначе. Модель mT5-base формирует короткий и в целом корректный набор предсказаний, но его охват остаётся ограниченным. Модель mBART-large действует сходным образом, однако её предсказания оказываются более ёмкими и точными: она возвращает немногочисленные, но хорошо согласованные с содержанием аннотации ключевые слова.

Базовая модель `ru-mbart-summ` и её модификации позволяют увидеть более сложные эффекты. В исходной конфигурации возникает семантическое искажение во фразе *управление материально-техническими учреждениями*, где нарушается смысловая связь между компонентами исходного текста. Локальная псевдоразметка делает предсказания устойчивее, однако по-прежнему не приводит к появлению содержательно более глубоких ключевых слов, не представленных в тексте в явном виде. Глобальная псевдоразметка несколько улучшает форму отдельных фраз, но в данном примере также не даёт заметного расширения множества отсутствующих в тексте ключевых слов.

Наиболее интересные изменения наблюдаются в конфигурациях `Generator` и `Generator-Ranker`. Конфигурация `Generator` начинает порождать более содержательные расширения, в частности *поддержка принятия решений*, что сближает её с эталонными ключевыми словами, не представленными в тексте в явном виде. Одновременно с этим появляются и явно нерелевантные элементы, такие как *методы многокритериального программирования*. После подключения модели семантического ранжирования часть шумовых генераций устраняется, однако полностью проблема не исчезает: в списке сохраняются слишком общие или тематически смещённые формулировки, например *информационные технологии*, *информационная система* и *измерительный комплекс*. Тем самым данный пример хорошо согласуется с общими количественными результатами: фильтрация уменьшает шум в экстрактивных методах, локальная псевдоразметка повышает устойчивость базовой генерации, а конфигурация `Generator-Ranker` с этапом семантического ранжирования расширяет множество потенциально релевантных ключевых слов, не представленных в тексте в явном виде, но делает это ценой появления дополнительных семантически близких, однако не всегда точных предсказаний.

Заключение

В работе предложен многоэтапный подход к автоматической генерации ключевых слов по русскоязычным научным аннотациям в условиях ограниченного объёма размеченных данных. Подход объединяет статистико-лингвистическую обработку автоматически выделенных фраз, построение псевдоразметки по неразмеченному корпусу, предварительное обучение генеративных моделей, контрастивное выделение фраз и последующее семантическое ранжирование.

Показано, что статистико-лингвистическая обработка выделенных фраз выполняет в системе не только функцию постобработки, но и играет более существенную роль. Именно очищенный список кандидатных фраз используется при построении псевдометок, выборе маскируемых фрагментов и формировании положительных и отрицательных примеров для контрастивного обучения. Тем самым фильтрация выступает как общий механизм организации обучающих данных на нескольких этапах модели.

Экспериментальные результаты показали, что применение статистико-лингвистического фильтра устойчиво улучшает качество экстрактивных методов, а предварительное обучение на локальной псевдоразметке с маскированием обеспечивает наилучший прирост по основной семантической метрике BERTScore@10 среди сопоставимых конфигураций. В свою очередь, конфигурация Generator-Ranker оказалась особенно полезной для компоненты ключевых слов, не представленных в тексте в явном виде, поскольку именно в этой части она дала наибольший прирост по диагностическим метрикам. Таким образом, различные компоненты системы решают разные подзадачи: локальная псевдоразметка прежде всего повышает общее качество генерации, тогда как

сочетание генератора и модели семантического ранжирования расширяет множество потенциально релевантных ключевых слов, не представленных в тексте в явном виде.

Отдельно следует отметить, что предварительное обучение на локальной псевдоразметке позволяет существенно сократить отставание компактной модели ru-mbart-summ от более крупной mBART-large по основным метрикам, особенно по BERTScore@10, что указывает на высокую эффективность такого способа использования неразмеченного корпуса.

Проведённый анализ показал, что глобальная псевдоразметка в текущей реализации не даёт преимуществ по сравнению с локальной схемой, что указывает на необходимость более строгого контроля качества внешних фраз, используемых для расширения обучающих данных.

К ограничениям работы следует отнести зависимость качества системы от набора эвристических правил, используемых в статистико-лингвистическом фильтре, а также сравнительную сложность полной архитектуры, включающей несколько взаимосвязанных этапов обучения и отбора. Кроме того, отдельного изучения требует влияние масштаба и качества неразмеченного корпуса на свойства псевдоразметки, а также переносимость предложенной схемы на другие научные домены и более широкие русскоязычные коллекции.

Перспективными направлениями дальнейшей работы являются улучшение глобальной схемы псевдоразметки, более точное управление качеством генерации ключевых слов, не представленных в тексте в явном виде, а также исследование облегчённых архитектур, позволяющих сохранить преимущества предложенного подхода при меньшей сложности итоговой системы.

ИСТОЧНИКИ ФИНАНСИРОВАНИЯ

Данная работа финансировалась за счёт средств бюджета университета. Никаких дополнительных грантов не привлекалось.

КОНФЛИКТ ИНТЕРЕСОВ

Авторы данной работы заявляют, что у них нет конфликта интересов.

СОБЛЮДЕНИЕ ЭТИЧЕСКИХ СТАНДАРТОВ

В данной работе отсутствуют исследования человека или животных.

Список литературы

1. T. Berg-Kirkpatrick, D. Burkett, and D. Klein, “An Empirical Investigation of Statistical Significance in NLP,” in Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jeju Island, Korea, 2012, pp. 995–1005.
2. A. Bougouin, F. Boudin, and B. Daille, “TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction,” in Proceedings of the Sixth International Joint Conference on Natural Language Processing, Nagoya, Japan, 2013, pp. 543–551.
3. R. Campos, V. Mangaravite, A. Pasquali, A. M. Jorge, C. Nunes, and A. Jatowt, “YAKE! Keyword Extraction from Single Documents using Multiple Local Features,” Information Sciences 509, 257–289 (2020). <https://doi.org/10.1016/j.ins.2019.09.013>
4. M. Choi, C. Gwak, S. Kim, S. Kim, and J. Choo, “SimCKP: Simple Contrastive Learning of Keyphrase Representations,” in Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, 2023, pp. 3003–3015. <https://doi.org/10.18653/v1/2023.findings-emnlp.199>
5. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, Minnesota, 2019, pp. 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
6. M. S. Dmitrieva and A. V. Zabrodin, “Автоматизация документооборота и бизнес-процессов с помощью ЕСМ-системы Docsvision,” Интеллектуальные технологии на транспорте, no. 2 (30), 25–31 (2022). <https://doi.org/10.24412/2413-2527-2022-230-25-31>
7. R. Dror, G. Baumer, S. Shlomov, and R. Reichart, “The Hitchhiker’s Guide to Testing Statistical Significance in Natural Language Processing,” in Proceedings of the 56th Annual

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 2018, pp. 1383–1392. <https://doi.org/10.18653/v1/P18-1128>

8. A. Glazkova and D. Morozov, “Exploring Fine-Tuned Generative Models for Keyphrase Selection: A Case Study for Russian,” in *Data Analytics and Management in Data Intensive Domains. DAMDID/RCDL 2024. Communications in Computer and Information Science* 2641 (Springer, Cham, 2026), pp. 98–111. https://doi.org/10.1007/978-3-032-03997-2_7

9. A. Glazkova, D. Morozov, and T. Garipov, “Key Algorithms for Keyphrase Generation: Instruction-Based LLMs for Russian Scientific Keyphrases,” in *Analysis of Images, Social Networks and Texts. AIST 2024. Lecture Notes in Computer Science* 15419 (Springer, Cham, 2025), pp. 107–119. https://doi.org/10.1007/978-3-031-88036-0_5

10. B. Kang and Y. Shin, “Improving Low-Resource Keyphrase Generation through Unsupervised Title Phrase Generation,” in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, Torino, Italia, 2024, pp. 8853–8865.

11. P. Koehn, “Statistical Significance Tests for Machine Translation Evaluation,” in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain, 2004, pp. 388–395.

12. K. Yu. Kolybanov, V. F. Korniyushko, M. M. Kulygina, S. N. Lobanov, and I. V. Khrapov, “Information Support of Monitoring of Material and Technical Facilities of State Educational Institutions,” *Trans. TSTU* 15 (3), 700–709 (2009).

13. C.-Y. Lin, “ROUGE: A Package for Automatic Evaluation of Summaries,” in *Text Summarization Branches Out*, Barcelona, Spain, 2004, pp. 74–81.

14. Y. A. Malkov and D. A. Yashunin, “Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42 (4), 824–836 (2020). <https://doi.org/10.1109/TPAMI.2018.2889473>

15. R. Mihalcea and P. Tarau, “TextRank: Bringing Order into Text,” in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain, 2004, pp. 404–411.

16. C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer,” *Journal of Machine Learning Research* 21 (140), 1–67 (2020).

17. G. Salton, A. Wong, and C. S. Yang, “A Vector Space Model for Automatic Indexing,” *Communications of the ACM* 18 (11), 613–620 (1975). <https://doi.org/10.1145/361219.361220>

18. Y. Tang, C. Tran, X. Li, P.-J. Chen, N. Goyal, V. Chaudhary, J. Gu, and A. Fan, “Multilingual Translation from Denoising Pre-Training,” in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Online, 2021, pp. 3450–3466. <https://doi.org/10.18653/v1/2021.findings-acl.304>

19. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention Is All You Need,” in *Advances in Neural Information Processing Systems* 30, 2017, pp. 5998–6008.

20. L. Wang, N. Yang, X. Huang, B. Jiao, L. Yang, D. Jiang, R. Majumder, and F. Wei, “Text Embeddings by Weakly-Supervised Contrastive Pre-training,” arXiv preprint arXiv:2212.03533 (2022).
21. T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, “BERTScore: Evaluating Text Generation with BERT,” in International Conference on Learning Representations, 2020.

Краткие биографии и фотографии всех авторов



Яушев Камиль, Yaushev Kamil, аспирант МГТУ им. Н.Э. Баумана. Область научных интересов: автоматическая обработка текстов, большие языковые модели, представление знаний.



Лукашевич Наталья Валентиновна, Loukachevitch Natalia Valentinovna, доктор технических наук, ведущий научный сотрудник НИВЦ МГУ имени М. В. Ломоносова, заведующая кафедрой факультета ВМК, профессор ВМК, филологического факультета МГУ.