Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра алгоритмических языков

Яковлева Светлана Андреевна

# Методы автоматического реферирования научных статей на русском языке

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

**Научный руководитель:**
к.ф-м.н.
Н.Э. Ефремова

Москва, 2024

## Аннотация

В данной магистерской диссертации проводится исследование применимости больших языковых моделей семейства T5 к задаче автоматического реферирования научных статей на русском языке. Были рассмотрены 2 модели размера Base и проведено дообучение модели rut5-small; ранее модель rut5-small для решения задачи автоматического реферирования не использовалась. Были проанализированы русскоязычные наборы данных, состоящие из пар *новость – ее реферат*, и посчитаны статистические характеристики для их последних версий. Дообучение модели проводилось на наборе данных Gazeta и созданных нами наборах данных RuSciText и RuArxiv, содержащих пары *научная статья – ее реферат*. Для оценки результатов работы моделей были вычислены 4 метрики: ROUGE, BLEU, METEOR и BERTScore, а также проведена экспертная оценка.

Анализ полученных результатов показал возможность использования моделей для автоматического реферирования научных статей на русском языке. В частности, значения метрик дообученной модели rut5-small получились выше, чем у моделей семейства GPT.

Результаты проведенного исследования выложены на GitHub RuSciTextSum[1].

[1] https://github.com/Svetych/RuSciTextSum

## Содержание

# 1. Введение

В настоящее время нам приходится работать с огромным количеством текстов различной природы. Как следствие, актуальной является разработка средств автоматического извлечения из текста всевозможной информации (его жанра, стиля, смысла, содержания, мнения автора и т.д.) и ее дальнейшая обработка [3]. В рамках данной магистерской диссертации рассмотрена задача **автоматического реферирования текстов** (Automatic Text Summarization, ATS), которая заключается в создании **реферата** – краткого содержания рассматриваемого текста, включающего основные идеи и ключевые моменты без избыточной информации [1].

Данная задача является одной из наиболее активно развивающихся в области **обработки естественного языка** (Natural language processing, NLP) и искусственного интеллекта в целом. Системы автоматического реферирования нашли широкое применение во многих прикладных задачах, таких как информационный поиск, автоматизированная журналистика, анализ социальных медиа и других [4, 31].

Существующие подходы к решению задачи автоматического реферирования принято разделять на **экстрагирующие** (extractive), **абстрагирующие** (abstractive) и **гибридные** (hybrid) [13]. Суть методов экстрагирующего подхода заключается в извлечении **квазиреферата,** или **экстракта** – наиболее важных фраз, предложений или абзацев из рассматриваемого текста. Рефераты, составленные таким образом, страдают от несвязности и избыточности, поскольку, как правило, представляют собой набор наиболее высоко оцененных предложений текста [3, 13]. К достоинствам данных методов относят простоту их реализации. Схема работы экстрагирующих систем [31] приведена на Рисунке 1.

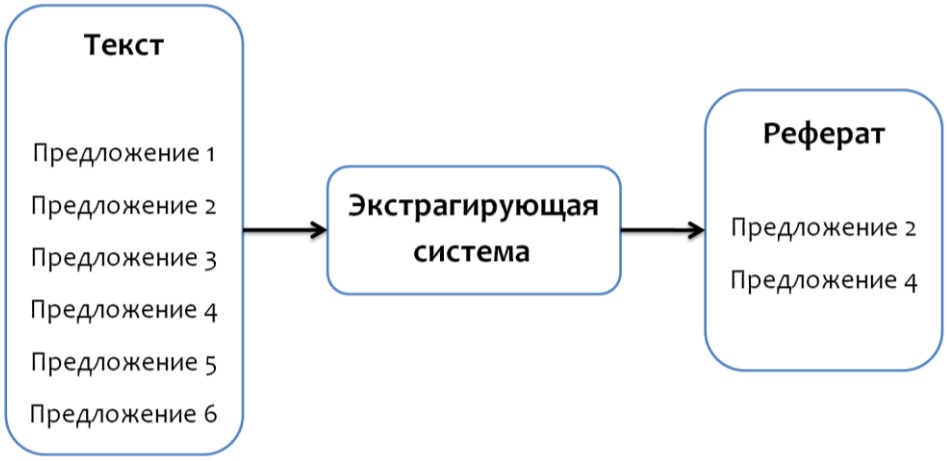



Рисунок 1: Схема работы систем экстрагирующего подхода

Абстрагирующий подход, также называемый *абстрактным*, *генерирующим* или *глубинным*, заключается в порождении нового текста из исходного с помощью **языковой модели**, являющейся, по сути, вероятностным распределением по последовательностям слов [31]. В последнее время особенно активно развиваются методы, использующие **нейронные, или большие языковые модели** (Large Language Models, LLM, НЯМ). Они представляют собой нейронные сети с большим количеством весов. Текст, полученный абстрагирующими методами, приближен к рефератам, созданным человеком, поскольку является более связным и при этом включает в себя основную информацию из исходного документа. Важно отметить, что абстрактные рефераты состоят из терминов, фраз и предложений, не обязательно входивших в исходный текст [3, 13]. Схема работы абстрагирующих систем приведена на Рисунке 2.

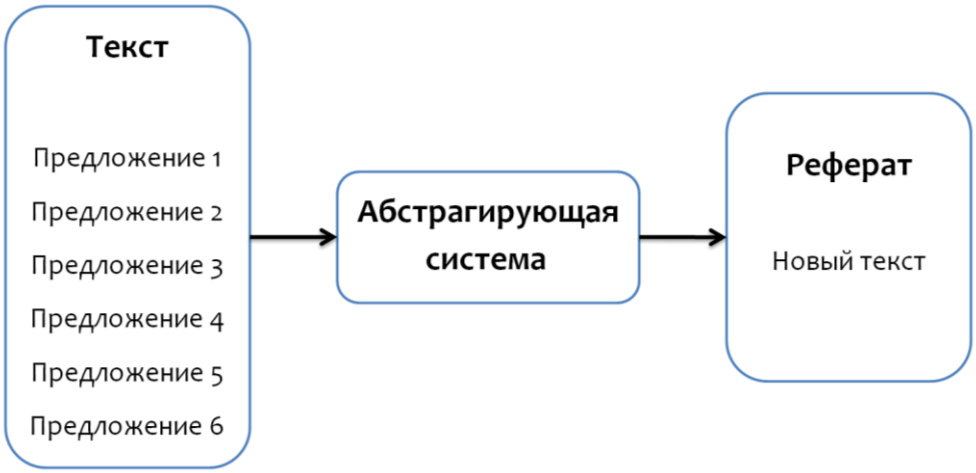



Рисунок 2: Схема работы систем абстрагирующего подхода

В [3, 39] отмечается, что, несмотря на успехи использования для реферирования LLM, из-за требований согласованности и плавности (*fluency* – свобода/плавность) изложения абстрактное реферирование в чистом виде все же требует высокого уровня семантического понимания текста, что выходит за рамки возможностей современных моделей. Поэтому данная задача является не до конца решенной, а большинство систем остаются экстрагирующими.

Гибридный подход основан на принципах обоих подходов и, как следствие, обладает их достоинствами и недостатками. Гибридных систем существует не так много (как правило, в пример приводят две из них [39]: SUMMARIST [19] и SumItUp [7]), поэтому в исследованиях часто этот подход отдельно не рассматривается.

В данный момент автоматическое реферирование активно развивается, но большинство исследований все еще проводится для английского языка. Причем рассматриваются различные тексты: новости, книги, медицинские документы, научные

статьи и т.д. [13]. Аналогичные исследования для других языков, в частности для русского, сталкиваются с рядом проблем, и основная из них – нехватка размеченных наборов данных из текстов различных тематик. Например, у нас в стране в первую очередь развивалось реферирование новостных статей, поэтому все существующие в открытом доступе русскоязычные датасеты состоят именно из них, т.е. содержат пары *новостная статья – ее реферат*.

Помимо новостных статей особый интерес в области автоматического реферирования всегда представляли научно-технические тексты [5]. Система, способная генерировать рефераты к научным статьям, была бы полезна как для читателей с целью быстрого получения представления о читаемой статье, так и для авторов в качестве инструмента для создания *аннотации*, т.е. более короткого варианта реферата к написанному тексту. Сейчас в открытом доступе не существует систем, позволяющих получать хорошие рефераты научных статей.

## 2. Постановка задачи

Цель данной магистерской диссертации – провести исследование методов, демонстрирующих наилучшие результаты при решении задачи автоматического реферирования новостей на русском языке, и проверить их применимость к задаче реферирования научных статей. Для достижения данной цели было необходимо решить следующие задачи:

1. Изучить современные методы абстрактного реферирования и выбрать из них, демонстрирующие наилучшие результаты для русского языка.
2. Исследовать русскоязычные наборы данных для задачи автоматического реферирования.
3. Разработать программное средство, реализующее выбранные методы.
4. Провести экспериментальное исследование работы реализованных методов и проанализировать полученные результаты.

# 3. Исследование современных методов автоматического реферирования

## 3.1. Методы абстрактного подхода

В [3] выделяются следующие методы абстрактного (абстрагирующего, генерирующего) подхода:

- Сжатие предложений (Sentence Compression);
- Слияние информации (Information Fusion);
- Упорядочивание информации (Information Ordering);
- Нейронные сети:
  - Рекуррентные нейронные сети (RNN);
  - Трансформеры (Transformers).

Методы *сжатия предложений* удаляют несущественные фразы из исходного текста. Эти методы можно разделить на методы *на основе правил*, использующие лингвистические знания для определения несущественных фраз, и *статистические* методы, строящие с помощью вероятностных КС-грамматик деревья для определения предложений, которые можно удалить.

Методы *слияния информации* объединяют информацию из нескольких предложений в одно, т.е. задача реферирования сводится к задаче поиска такого объединения двух предложений, которое будет наилучшим образом передавать информацию, содержащуюся в них.

Методы *упорядочивания информации* предназначены для создания реферата по нескольким документам со схожей тематикой. Для этого строится граф, в котором каждая вершина представляет собой тематический кластер. Ребро помещается между двумя вершинами, если в документе одна тема предшествует другой. Порядок тем в различных документах используется для определения того, какие темы должны предшествовать друг другу в итоговом реферате. Каждый кластер сводится к одному предложению с помощью экстрагирования (выбора одного репрезентативного предложения), либо с помощью перефразирования.

В [3] также говорится о схожести методов решения задачи автоматического реферирования и задачи машинного перевода, из чего следует, что для получения реферата можно использовать *RNN* архитектуры нейронных сетей [26]; при этом входная последовательность будет длиннее, чем выходная. Отметим, что простое дообучение нейросетевых моделей машинного перевода невозможно, поскольку для задачи

автоматического реферирования размеры входных и выходных обучающих текстов намного больше, чем для задачи машинного перевода.

Эффективная обработка длинных входов представлена в сетях архитектуры ***Трансформер*** (Tramsformer) [35] (см. Рисунок 3). Трансформеры состоят из кодирующих элементов – стека **энкодеров**, и декодирующех элементов – стека **декодеров**. Ключевым элементом является слой **самовнимания** (self-attention), позволяющий учитывать при кодировании отдельного слова его контекст. Механизм многоголового самовнимания позволяет одновременно учитывать различные части входной последовательности.

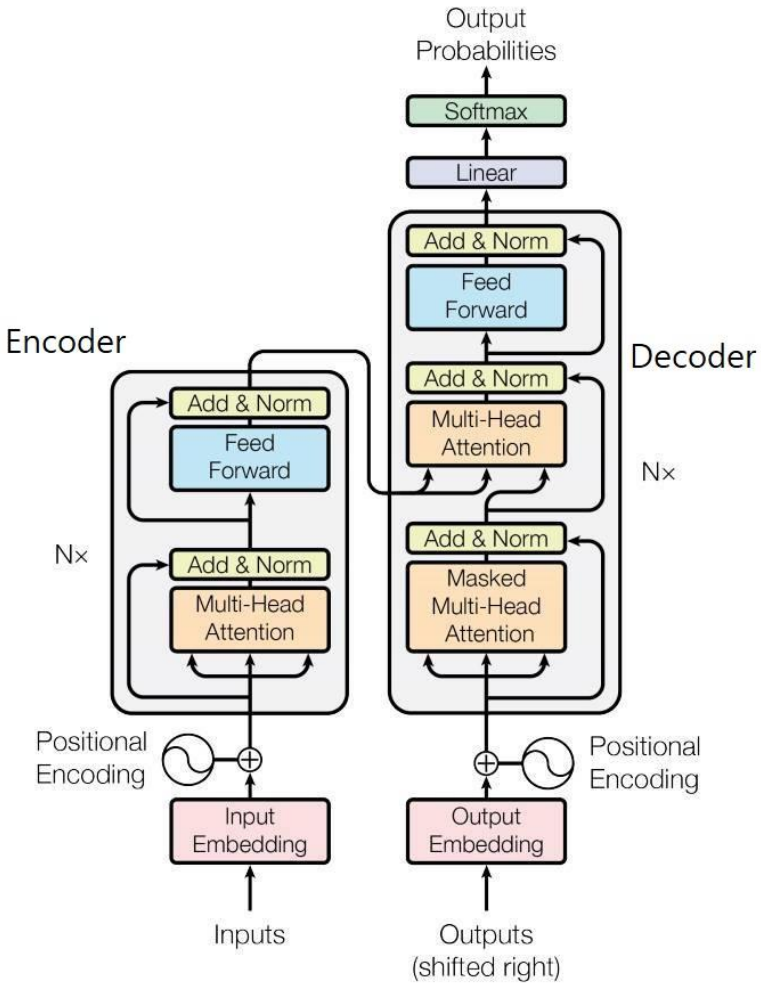



Рисунок 3: Схема архитектуры Transformer

Наиболее известными семействами языковых моделей, построенными на основе архитектуры Transformer, являются **BERT**, **GPT** и **T5**. Они показывают одни из самых лучших результатов в задачах обработки естественного языка, в том числе и в задаче автоматического реферирования. Отметим, что существуют модели, предобученные именно на русскоязычных текстах.

### 3.2. Большие языковые модели

Как говорилось ранее, большие языковые модели (LLM) – это нейронные сети с большим количеством весов, или параметров, способные предсказывать вероятность следующего слова по предыдущим. Для создания LLM необходимо обучить нейронную сеть на огромных объемах текстов. При этом она учится решать определенные задачи, например задачу *маскирования*, которая заключается в предсказании слов исходного текста, замененных на «маску» [12]. Затем LLM можно **дообучить** на специальных наборах данных для решения конкретной задачи NLP [24]. Так, для задачи реферирования используются наборы данных, состоящие из пар *текст – его реферат*.

Рассмотрим подробнее наиболее популярные на данный момент LLM.

**BERT** (Bidirectional Encoder Representations from Transformers) [12] – семейство языковых моделей появившееся в 2018 году. Типичная архитектура нейронной сети данного семейства представляет собой стек энкодеров Transformer. Одним из способов дообучения модели BERT для задачи автоматического реферирования является добавление декодера, который должен быть обучен с нуля.

**GPT** (Generative Pre-trained Transformer) [28] впервые появилась также в 2018 году и представляет собой стек декодеров Transformer. Модели данного семейства больше подходят для задачи генерации текста, поэтому их проще всего дообучить для задачи автоматического реферирования [3]. Для этого необходимо на вход нейронной сети подать исходный текст и его реферат, разделенные специальным токеном-разделителем. При этом модель учится достраивать часть последовательности, следующей после разделителя, в нашем случае – реферат.

**T5** (Text-to-Text Transfer Transformer) [29] появилась в 2020 году и представляет собой полноценный Transformer (содержит и стек энкодеров, и стек декодеров). Изначально модель обучалась для многих задач NLP; вид решаемой задачи указывался в префиксе подаваемой в нейронную сеть последовательности (см. Рисунок 4). К примеру, префикс `translate English to German:` соответствовал задаче машинного перевода с английского на немецкий, а префикс `summarize:` – задаче автоматического реферирования. Однако существующие многозадачные модели T5 не обучались для решения задачи автоматического реферирования именно на русском языке. Чтобы их дообучить, на вход модели необходимо подавать исходный запрос (текст) и реферат в качестве метки к нему.

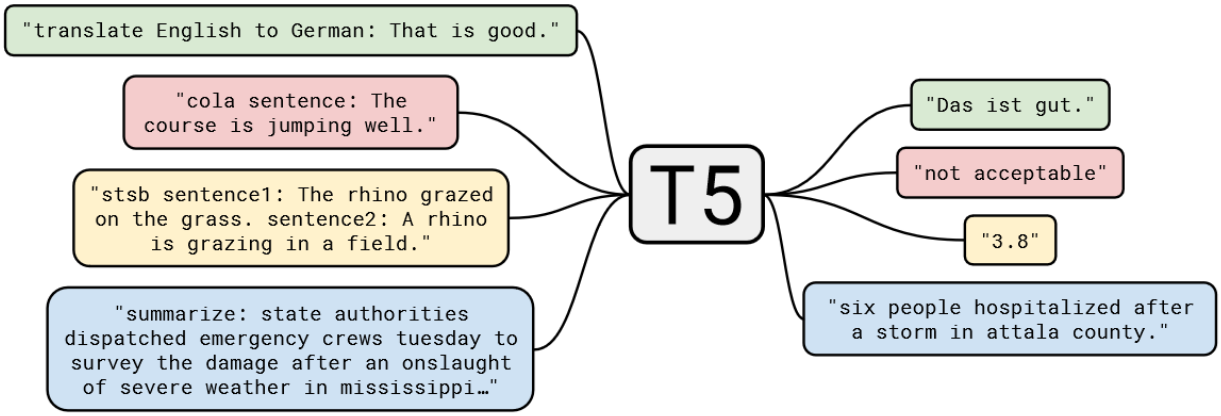



Рисунок 4: Задачи, для которых обучались модели семейства T5

В работе [3] отдельно рассматривается особый вид обучения – **самообучение** (Self-Supervised Training). Для задачи автоматического реферирования была специально создана нейронная сеть **PEGASUS** (Pre-training with Extracted Gap-sentences for Abstractive Summarization) [40], также имеющая архитектуру Transformer. Сеть самообучалась на задаче *генерации пропущенных предложений* (Gap Sentences Generation, GSG), для которой не требуются предварительно размеченные данные. Суть обучения состояла в том, что из исходного текста автоматически убирались важные предложения, которые модель должна была восстановить, и текст подавался на вход модели. Модель училась выявлять семантические связи между предложениями, определять, какие из них являются важными, и генерировать пропущенные фрагменты текста. На момент выхода статьи (в 2020 году) PEGASUS демонстрировала лучшие результаты среди всех существовавших методов реферирования. Дополнительно авторами была создана модель PEGASUS-X [30], предназначенная для реферирования длинных входных текстов объемом до 16 000 токенов.

Модель PEGASUS хорошо работает при ограниченных ресурсах, а именно при нехватке размеченных данных. Авторы отмечают, что для дообучения модели достаточно 1 000 примеров, чтобы достигнуть результатов, близких к результатам, получаемым ведущими моделями того времени [40].

PEGASUS и PEGASUS-X отлично подходят для решения рассматриваемой в данной магистерской работе задачи, т.к. тексты научных статей, как правило, довольно объемные. Однако для русского языка не существует предобученных моделей данного семейства. Обучение же этих моделей с нуля довольно ресурсозатратно (модель PEGASUS имеет 568 млн, а модель PEGASUS-X 569 млн параметров) и требует огромного объема текстовых данных, поэтому в рамках данной работы было решено отказаться от их рассмотрения.

## 3.3. Наборы данных для автоматического реферирования

Как уже было сказано, дообучение языковых моделей для задачи автоматического реферирования требует использования размеченных наборов данных, или датасетов. **Разметка** заключается в составлении пар вида *текст – его реферат*. **Эталонные**, или «золотые» рефераты чаще всего создаются экспертами вручную или с использованием вспомогательных программных средств, что само по себе является достаточно трудоемким процессом.

### 3.3.1. Русскоязычные наборы данных

В открытом доступе для русского языка существует не так много наборов данных, на которых можно обучать LLM для решения задачи автоматического реферирования, а именно только новостной датасет Gazeta [15, 16], состоящий из 74 126 пар *статья – реферат*, собранных с ресурса Gazeta.ru.

Для ***мультиязыкового реферирования*** (языковая модель обрабатывает тексты на разных языках) существуют специальные мультиязычные корпуса. Русскоязычные части таких корпусов можно использовать для обучения НЯМ для решения поставленной в данной диссертации задачи.

Существуют следующие датасеты для мультиязыкового реферирования:

- **MLSUM** (MultiLingual SUMmarization) [33], состоящий из 27 063 пар.
- **XL-Sum** [18, 38], состоящий из 77 803 пар.
- **WikiLingua** [23, 36], состоящий из 52 928 пар.

В исследовании [17] приводятся статистические характеристики корпусов Gazeta, MLSUM и XL-Sum. Данные рассчитаны для токенов, полученных с помощью токенизатора *razdel* [32]. Дополнительно в работе [17] отмечается, что лишь 54 текста из датасета MLSUM (0,2%) имеют длину более 5 000 токенов. В датасете XL-Sum лишь 96 текстов (0,15%) длиной более 5 000 токенов, 5 рефератов длиной менее 3 токенов и 23 реферата длиной более 100 токенов. В целом получается, что очень длинные и очень короткие тексты нетипичны для имеющихся наборов данных.

По аналогии с исследованием [17] нами было принято решение провести исследование статистических характеристик набора WikiLingua. Его результаты представлены в Разделе 4.3; там же содержится более подробная информация по наборам Gazeta, MLSUM и XL-Sum.

### 3.3.2. Англоязычные наборы данных

Что касается английского языка, то в открытом доступе существует достаточно много наборов данных, содержащих тексты различных жанров и стилей. Например, датасет BookSum [9, 22] специально создавался для обработки текстов книг, которые являются объемными и сложными по структуре; в нем три варианта эталонных рефератов: для абзацев, для глав и для всего произведения.

Для реферирования научных статей существуют следующие наборы данных:

- **Arxiv** [8], состоящий из 215 913 пар.
- **PubMed** [8], состоящий из 133 215 пар.
- **Sci Lay** [10], состоящий из 43 790 пар.

В Таблице 1 представлены статистические характеристики по этим датасетам. Статистики для датасета Sci Lay была посчитана нами с помощью токенизатора[2] для rut5-small.

Интересно, что набор данных Sci Lay, состоящий из медицинских статей, содержит 2 варианта эталонных рефератов: реферат с использованием научной лексики и более общий реферат. Также отметим, что тексты для датасетов Arxiv и PubMed собраны с открытых ресурсов arXiv.org и PubMed.com соответственно.

Таблица 1: Статистика по англоязычным датасетам

| Датасет | Выбока | Размер | Данные | Средняя длина в токенах |
|---|---|---|---|---|
| Arxiv[3] | train | 203 037 | текст | 6 038 |
| | | | реферат | 299 |
| | validation | 6 436 | текст | 5 894 |
| | | | реферат | 172 |
| | test | 6 440 | текст | 5 905 |
| | | | реферат | 174 |
| PubMed[4] | train | 119 924 | текст | 3 043 |
| | | | реферат | 215 |
| | validation | 6 633 | текст | 3 111 |
| | | | реферат | 216 |
| | test | 6 658 | текст | 3 092 |
| | | | реферат | 219 |
| Sci Lay | train | 35 026 | текст | 18 300 |
| | | | реферат | 367 |
| | validation | 4 380 | текст | 18 261 |
| | | | реферат | 367 |

[2] Токенизатор для rut5-small: https://huggingface.co/cointegrated/rut5-small
[3] Датасет Arxiv и статистика по нему: https://huggingface.co/datasets/ccdv/arxiv-summarization
[4] Датасет PubMed и статистика по нему: https://huggingface.co/datasets/ccdv/pubmed-summarization

| | | | | |
|---|---|---|---|---|
| Sci Lay | test | 4 384 | текст | 18 410 |
| | | | реферат | 368 |

## 3.4. Оценка качества автоматического реферирования

### 3.4.1. Метрики оценки качества реферата

Для оценки качества систем автоматического реферирования текстов используются различные метрики, оценивающие схожесть сгенерированного реферата с эталонным. Наиболее популярными метриками являются (перечислим их в порядке увеличения корреляции с человеческим мнением о качестве сгенерированного реферата):

- **ROUGE** (Recall-Oriented Understudy for Gisting Evaluation) [25];
- **BLEU** (Bilingual Evaluation Understudy) [27];
- **METEOR** (Metric for Evaluation of Translation with Explicit ORdering) [6];
- **BERTScore** [42].

ROUGE и BERTScore позволяют вычислить *точность* (precision), *полноту* (recall) и *f-меру*, в то время, как BLEU и METEOR являются числовыми показателями качества.

С момента создания метрика ROUGE считается стандартом измерения качества в задаче реферирования [41]. Метрики BLEU и METEOR предназначены для оценки качества систем машинного перевода, но также используются для оценки сгенерированных рефератов. BERTScore использует в своей основе контекстуальные эмбеддинги из предобученной модели BERT, что позволяет ей учитывать семантику оцениваемого текста. В Приложении В метрики рассмотрены более подробно, для каждой приведены формулы их вычисления.

Стоит отметить, что современные исследования призывают учитывать при оценке качества моделей *чувствительность* метрик. Так, метрика ROUGE считается недостаточно чувствительной, поскольку малые изменения ее значений не могут говорить о действительном улучшении качества генерации [11]. Однако при сильной разнице в значениях метрик можно отличить более качественную модель от менее качественной.

### 3.4.2. Показатели метрик для русскоязычных моделей

В исследовании [17] сравнивалась работа экстрагирующих и абстрагирующих систем автоматического реферирования для русского языка. В качестве НЯМ были взяты ruT5-large (737 млн параметров), mBART (1 123 млн параметров), ruT5-base (222 млн параметров), ruGPT3Large (762 млн параметров), ruGPT3Small (117 млн параметров). Для оценки качества использовались четыре рассмотренные выше метрики и датасеты Gazeta,

MLSUM и XL-Sum. Результаты исследования приведены в Таблице 2, для ROUGE и BERTScore указывается f-мера. Лучше всего себя показала модель ruT5-large.

Таблица 2: Результаты оценки НЯМ для русского языка

| Датасет | Модель | ROUGE-1 | ROUGE-2 | ROUGE-L | BLUE | METEOR | BERTScore |
|---|---|---|---|---|---|---|---|
| Gazeta | mBART | 31.55 | 13.54 | 28.22 | **11.19** | **34.09** | 56.56 |
| | ruT5-base | 30.45 | 12.63 | 27.41 | 9.54 | 28.69 | 56.35 |
| | ruT5-large | **32.45** | **13.97** | **29.24** | 10.88 | 31.21 | **57.73** |
| | ruGPT3Small | 18.84 | 4.06 | 16.68 | 3.13 | 18.70 | 44.06 |
| | ruGPT3Large | 23.45 | 6.45 | 20.73 | 4.93 | 23.77 | 47.76 |
| MLSUM | mBART | 11.48 | 1.95 | 10.26 | 1.49 | 10.52 | 37.89 |
| | ruT5-base | 12.35 | 1.86 | 11.22 | 1.58 | 9.68 | 38.67 |
| | ruT5-large | **14.06** | **2.86** | **12.69** | **2.81** | **11.84** | **39.92** |
| | ruGPT3Small | 9.14 | 0.60 | 8.13 | 0.40 | 6.66 | 34.27 |
| | ruGPT3Large | 9.36 | 0.99 | 8.17 | 0.73 | 7.44 | 35.00 |
| XL-Sum | mBART | 26.47 | 10.95 | 22.67 | 7.51 | 27.16 | 54.24 |
| | ruT5-base | 26.52 | 10.67 | 22.79 | 6.58 | 25.35 | 52.89 |
| | ruT5-large | **28.42** | **11.98** | **24.41** | **7.93** | **28.31** | **56.06** |
| | ruGPT3Small | 16.19 | 3.28 | 13.68 | 2.25 | 15.94 | 40.12 |
| | ruGPT3Large | 19.37 | 5.17 | 16.48 | 3.74 | 19.63 | 42.74 |

Из таблицы видно, что результаты работы моделей одного семейства с разным количеством параметров (версии Large и Small/Base) не так сильно отличаются, как можно было бы ожидать – всего лишь на 1-2 единицы. Низкие результаты для набора данных MLSUM объясняются авторами исследования двумя причинами:

1. MLSUM в 2 раза меньше датасета Gazeta и в 3 раза меньше XL-Sum.
2. Средняя длина рефератов MLSUM меньше, чем у двух других датасетов.

Также в [17] отмечаются некоторые особенности работы рассматриваемых нейронных языковых моделей. mBART повторяла части предложений из оригинального текста в сгенерированных рефератах. Из остальных моделей у нее меньше всех доля новых n-грамм, что говорит о низкой абстрактности. Наибольшая доля новых n-грамм у семейства GPT. С другой стороны, у его моделей замечено больше всего ошибок – несоответствий исходному тексту. У моделей семейства T5 ошибок намного меньше, а абстрактность больше, чем у mBART.

### 3.5. Выводы

Для достижения цели, поставленной в данной диссертации, было решено исследовать работу нейронных языковых моделей, поскольку для задачи автоматического реферирования они показывают лучшие результатов. Для исследования было выбрано семейство моделей T5, демонстрирующих лучшие результаты для русского языка.

Реферирование документов, принадлежащих конкретной прикладной области, называется *предметно-специфическим* [13]. В ряде исследований отмечается, что обучение моделей автоматического реферирования для конкретной прикладной области улучшает качество сгенерированных рефератов по сравнению с системами, которые не нацелены на обработку каких-то определенных текстов [3]. Это обусловлено наличием специфической лексики и терминологии, которую модель запоминает во время обучения.

Рассмотренные нами в подразделе 3.3.1 наборы данных на русском языке содержат в себе новостные статьи. В открытом доступе русскоязычных наборов данных, состоящих из научно-технических текстов и их рефератов, на данный момент не существует. В связи с этим проблему нехватки данных для дообучения модели было предложено решить тремя способами:

- использовать имеющийся русскоязычный датасет;
- вручную разметить набор данных небольшого размера;
- автоматически перевести с другого языка часть существующего набора данных нужной тематики.

# 4. Создание наборов данных из научных статей на русском языке

Как уже было сказано, для настройки языковой модели на решение задачи автоматического реферирования научных статей необходимо дообучить ее на наборе данных, содержащем именно научно-технические тексты и их рефераты. Однако на данный момент для русского языка в открытом доступе таких наборов не существует, поэтому нами было принято решение создать их самостоятельно.

В работе [34] говорится, что качество набора данных, на котором происходит обучение, оказывает существенное влияние на результаты работы итоговой модели. Источником проблем может быть, например, некачественный реферат: несвязанный с исходным документом, внезапно обрывающийся, содержащий лишние символы или лишнюю информацию, полностью состоящий из фраз документа. Поскольку при полностью автоматической обработке и разметке текстов вероятность возникновения подобных проблем очень велика, процессу ручного контроля над созданием датасетов в данной работе уделено особое внимание.

## 4.1. Набор данных RuSciText

Сначала нами был вручную размечен набор данных из 100 научных статей, который далее будем называть **RuSciText**.

Изначально в нашем распоряжении имелась текстовая коллекция из 1 651 документов, содержащая статьи в формате pdf и их версии в формате txt, полученные путем автоматического конвертирования. Статьи относятся к различным научным областям: компьютерная лингвистика, искусственный интеллект, медицина, вычислительная математика, компьютерная разработка и др. После предварительного анализа нами были обнаружены следующие особенности:

1. В коллекции встречаются статьи с русскими заголовками и аннотациями, основной текст которых написан на английском языке. Такие тексты – 238 файлов – были объединены нами в группу *en*.
2. Некоторые документы содержат в себе исключительно аннотации, приложения и другие разделы научных сборников, не относящиеся к статьям. Такие тексты – 245 файлов – были объединены в группу *inappropriate*.
3. Имеются статьи, текст которых состоит из нескольких колонок, что требует дополнительной обработки. Такие тексты – 35 файлов – были объединены в группу *two*.
4. Файлы, содержащие сборники статей, были объединены в группу *compendium* – 32 файлов.

5. Поскольку научные статьи, как правило, достаточно длинные (более 2-х страниц текста), короткие тексты тезисов были объединены в отдельную группу *thesis* – 18 файлов.
6. Не все документы были без ошибок преобразованы из pdf-формата в читаемый txt-текст. Такие документы были удалены.

Остальные тексты были объединены в 3 основные группы по размеру исходного текста (файл размером 50 КБ соответствует примерно 7 страницам текста или примерно 3 500 токенам):

- *небольшие* (размер txt-файла менее 50КБ): 447 файлов;
- *средние* (размер txt-файла 50-100КБ): 180 файлов;
- *большие* (размер txt-файла больше 100КБ): 24 файла.

На данный момент не существует универсальных инструментов быстрой автоматической очистки текстов, полученных при конвертации в txt-формат. Поэтому для автоматизации процесса очистки и предобработки текстов нами была написана собственная программа на языке *Python 3*. С ее помощью были осуществлены следующие преобразования:

1. Приведение документов к единому виду:
   - Каждая пара *статья – реферат* представлена в отдельном файле в формате txt и кодировке utf-8.
   - Название статьи, ее текст и эталонный реферат разделены пустой строкой и расположены именно в таком порядке.
2. Удаление лишних и замена неправильно распознанных символов.
3. Удаление формул, таблиц, рисунков, схем, номеров страниц:
   - Если нетекстовый элемент был частью предложения, то он заменялся своим номером в статье (если номер был присвоен).
   - Иначе элемент удалялся из текста с сохранением связности.
4. Удаление разделов:
   - Аннотация;
   - Ключевые слова;
   - Список литературы;
   - Названия конференций, сборников и издательств.
5. Удаление информации об авторах.

Затем для проведения дальнейших экспериментов вручную были размечены 100 текстов *небольшого* размера (до 50 Кб). Особое внимание уделялось тому, чтобы в эталонных рефератах были охвачены все основные аспекты оригинальной статьи, такие

как проблема, задача исследования, используемые методы и результаты работы. За основу брались оригинальные аннотации к статьям, если они были. Обработанные тексты и полученный набор данных доступны на Яндекс-диске[5].

### 4.2. Набор данных RuArxiv

Для получения второго набора данных было решено осуществить автоматический перевод статей из англоязычного датасета. Был выбран датасет **Arxiv**, содержащий научные тексты из разных областей (а, например, не только из медицины). Далее будем называть полученные набор данных **RuArxiv**.

При анализе датасета Arxiv были выявлены следующие особенности:

1. Тексты содержат в себе нетекстовые элементы исходной статьи, переведенные в разметку, такие как формулы, рисунки, таблицы, математические символы, ссылки и сноски и т.д. Зачастую эти элементы разметки состоят из большого количества неинформативных символов.
2. Тексты содержат лишние пробельные символы, которые, предполагаемо, остались после процесса конвертации.

Для автоматизации формирования набора данных была написана программа на языке *Python 3*; перевод датасета Arxiv велся с использованием технологий Yandex Cloud[6]. С помощью написанной программы была проделана следующая работа:

1. Предварительная обработка исходных текстов:
    - Отбор статей:
        - Текст и статья не пустые;
        - Длина текста статьи более 50 символов, а длина реферата более 10 символов.
    - Ручная проверка, является ли текст научной статьей (например, встречались тексты, состоящие только из списка литературы).
    - Ручное удаление разделов acknowledgments и references, сведений о конференции и авторах, нетекстовых элементов разметки исходной статьи.
    - Преобразование исходного текста с целью экономии символов (в порядке их применения):
        - Замена символов ‘\n’ + пробел на пустую строку;
        - Удаление пробелов перед ‘.’ ;
        - Удаление пробелов перед ‘,’ ;

[5] Набор данных RuSciText: https://disk.yandex.ru/d/UW1_0JASBxJUbw
[6] Yandex Translate: https://cloud.yandex.ru/ru/docs/translate/

- Замена двойных пробелов на одинарные.

2. Перевод текстов:
   - Разбиение текста на части по 10 000 символов.
   - Обращение к API Yandex Translate.
   - Соединение переведенных частей текста в один документ.
3. Приведение документов к единому виду:
   - Каждая пара *статья – реферат* представлена в отдельном файле в формате txt и кодировке utf-8.
   - Статья и эталонный реферат разделены пустой строкой и расположены именно в таком порядке.

Во время анализа получившегося набора данных в некоторых случаях статья и реферат к ней менялись местами, поскольку они были перепутаны в исходном датасете. Процент сжатия – см. Раздел 4.3 – для таких случаев сильно превышал 100%.

Таким образом на русский язык было переведено 400 статей из датасета Arxiv. Полученный набор данных, а также файлы с результатами каждого этапа обработки выложены на Яндекс-диске[7].

### 4.3. Статистическое исследование наборов данных

По аналогии с исследованием [17] нами был проведен статистический анализ собранных наборов данных. Дополнительно был вычислен **процент сжатия информации** – отношение длины реферата в токенах к длине его исходного текста. Такие же характеристики были вычислены для последних доступных версий датасетов Gazeta, MLSUM, XL-Sum и WikiLingua.

Анализ результатов начнем с новостных наборов данных. В Таблице 3 представлена статистика по датасету Gazeta, в Таблице 4 – по WikiLingua. Результаты, полученные для новостных наборов MLSUM и XL-Sum, см. в Приложении А.

Из таблиц и графиков видно, что длины большинства текстов находятся в следующих диапазонах и имеют следующие распределения:

- **Gazeta**: 300-1400 токенов, нормальное распределение по длинам.
- **MLSUM**: 55-4 500 токенов, основная часть длиной до 2 000 токенов, логнормальное распределение.
- **XL-Sum**: 19-1500 токенов, экспоненциальное распределение.
- **WikiLingua**: 1-1500 токенов, логнормальное распределение.

[7] Набор данных RuArxiv: https://disk.yandex.ru/d/c_gE8O4fpPj1_w

Из этого можно сделать вывод, что очень длинные тексты являются для новостных наборов скорее исключением.

Таблица 3: Статистика по датасету Gazeta

| Выборки | Размер | Данные | Длина в токенах | | |
|---|---|---|---|---|---|
| | | | min | max | mean |
| train | 60 964 (82.2%) | текст | 28 | 1 500 | 766,88 |
| | | реферат | 15 | 85 | 49,57 |
| | | *процент сжатия* | *1,11 %* | *178,57 %* | *6,83 %* |
| validation | 6 369 (8.6%) | текст | 344 | 1 500 | 723,95 |
| | | реферат | 15 | 85 | 53,09 |
| | | *процент сжатия* | *1,15 %* | *19,6 %* | *7.72 %* |
| test | 6 793 (9.2%) | текст | 246 | 1 500 | 732,04 |
| | | реферат | 15 | 85 | 53,91 |
| | | *процент сжатия* | *1,2 %* | *23,68 %* | *7,83 %* |

Таблица 4: Статистика по датасету WikiLingua

| Выборки | Размер | Данные | Непустые | Длина в токенах | | |
|---|---|---|---|---|---|---|
| | | | | min | max | mean |
| train | 37 029 (70%) | текст | 37 024 | 1 | 4 216 | 378,3 |
| | | реферат | 37 028 | 2 | 917 | 38,3 |
| | | *сжатие* | | *0,4 %* | *14 200 %* | *20,72 %* |
| validation | 5 289 (10%) | текст | 5 286 | 3 | 4 697 | 383,1 |
| | | реферат | 5 288 | 3 | 517 | 39,9 |
| | | *сжатие* | | *0,17 %* | *1 533,33 %* | *23,31 %* |
| test | 10 581 (20%) | текст | 10 577 | 4 | 5 354 | 373,7 |
| | | реферат | 10 580 | 2 | 464 | 39,2 |
| | | *сжатие* | | *0,28 %* | *2 250 %* | *23,84 %* |

Что касается датасета WikiLingua, отдельно отметим, что количество текстов, длина которых больше 3 000 токенов, во всем наборе данных всего 8 (0,015%), менее 5 токенов – 31 (0,059%), количество рефератов длиной меньше 5 – 411 (0,777%). Однако при незначительности количества таких данных при расчете процента сжатия они вносят сильную погрешность в минимальные и максимальные значения.

Кроме того, из Таблицы 4 видно, что датасет WikiLingua требует дополнительной обработки, поскольку в разделении по выборкам, предложенном создателями [36], были обнаружены пропуски, и итоговое количество статей не совпадает с заявленным. Анализ сжатия информации говорит о том, что во многих парах текст был перенесен с ошибками, либо же перепутан местами с рефератом к нему, поскольку длина реферата оказывалась больше длины статьи.

Таким образом, анализ существующих наборов данных показал, что эталонный реферат чаще всего состоит из одного-двух предложений и существенно короче текста новостной статьи, что говорит о высокой степени сжатия информации из исходного текста. Можно сказать, что они имеют *индикативный* характер, т.е. рефераты содержат только общую идею об исходном тексте [13]. Для научных статей предпочтительней, чтобы реферат включал в себя всю важную информацию о тексте, которую, как правило, сложно уместить в одном предложении. Рефераты, содержащие важную информацию и идеи оригинального текста, охватывающие его основное содержание без подробностей, будем называть **информативными** [13]. В процессе разметки набора RuSciText мы особо следили за тем, чтобы рефераты получились именно такими.

Подробная статистика по сформированным нами наборам RuSciText и RuArxiv приведена в Таблице 5. Графики распределения размеченных текстов по их длинам представлены в Приложении А.

Таблица 5: Статистика по созданным наборам данных

| Датасет | Размер | Данные | Длина в токенах | | |
|---|---|---|---|---|---|
| | | | min | max | mean |
| RuSciText | 100 | текст | 781 | 3 263 | 1 928,8 |
| | | реферат | 47 | 358 | 126,54 |
| | | *процент сжатия* | *1,81 %* | *20,24 %* | *7,00 %* |
| RuArxiv | 400 | текст | 448 | 54 050 | 6 174,1 |
| | | реферат | 17 | 2 291 | 187,71 |
| | | *процент сжатия* | *0,22 %* | *74,81 %* | *4,84 %* |

Для набора данных RuSciText средняя длина в токенах текстов, обработанных токенизатором *razdel* [32], оказалась равна 1 928,8, а рефератов – 126,54. Следовательно, процент сжатия информации в полученном наборе данных в среднем выше, чем в новостных датасетах, что говорит о большей информативности эталонных рефератов. Тексты по длинам распределены почти равномерно.

Анализ набора RuArxiv показал, что количество текстов, длина которых больше 13 000 токенов, всего 27 (*0,07*%). В среднем длина эталонных рефератов находится в диапазоне от 25 до 500 токенов. Длины текстов в диапазоне до 10 000 токенов распределены нормально. Несмотря на то, что для части набора мы ранее меняли местами перепутанные статью и реферат к ней, максимальный процент сжатия все еще остается высоким (почти 75%). Это говорит об ошибках в оригинальном датасете, которые возникли при разметке документов.

# 5. Дообучение модели rut5-small для решения задачи автоматического реферирования

Напомним, что для достижения поставленной в данной диссертации цели по результатам анализа современных методов абстрактного реферирования для исследования нами было выбрано семейство моделей T5. При этом нами:

1. Была проанализирована работа дообученных моделей размера Base (244 млн параметров), доступ к которым является открытым [17]. Мы взяли модели **ruT5-base-absum**[8] и **rut5_base_sum_gazeta**[9], представленные с помощью средств платформы Hugging Face [20]:
   - Модель ruT5-base-absum основана на мультизадачной модели rut5-base-multitask[10], обученной, в том числе, на текстах английского и русского языков. Она была дообучена на датасетах Gazeta, XL-Sum, MLSUM и WikiLingua.
   - Модель rut5_base_sum_gazeta основана на русско-английской модели rut5-base[11] и была дообучена на задачу реферирования только на датасете Gazeta.
2. Была дообучена модель размера Small, а именно – модель **rut5-small**[12], основанная на мультиязычной модели mt5-small[13] (65 млн параметров). В задачах автоматического реферирования данная модель ранее не рассматривалась.

Целью нашего исследования также была проверка, насколько сильно модель с меньшим количеством параметров будет уступать по качеству моделям того же семейства с большим количеством параметров.

Анализ наборов данных показал, что в среднем длина эталонных рефератов для научных статей больше, чем для новостей. Также известно, что при обучении НЯМ возможно изменение параметра *max_length*, отвечающего за максимальную длину выхода. Однако изменением этого параметра невозможно увеличить размер генерируемого реферата, можно только ограничить его длину сверху. Следовательно, для того чтобы рефераты получались более длинными и информативными, необходимо, чтобы модель обучалась на соответствующих наборах данных.

[8] rut5-base-absum: https://huggingface.co/cointegrated/rut5-base-absum
[9] rut5_base_sum_gazeta: https://huggingface.co/IlyaGusev/rut5_base_sum_gazeta
[10] rut5-base-multitask: https://huggingface.co/cointegrated/rut5-base-multitask
[11] rut5-base: https://huggingface.co/cointegrated/rut5-base
[12] rut5-small: https://huggingface.co/cointegrated/rut5-small
[13] google/mt5-small: https://huggingface.co/google/mt5-small

Для полноценного дообучения моделей объема собранных нами наборов данных недостаточно; они подойдут именно для финальной настройки модели на решение задачи автоматического реферирования научных статей на русском языке. По этой причине для начального дообучения было решено использовать датасет Gazeta, имеющий длинные эталонные рефераты и довольно высокий процент сжатия. Кроме того, датасет Gazeta содержит большое количество текстов, длина которых не имеет явных выбросов (см. Приложение А), поэтому для него не требуется дополнительно отбирать тексты по длине.

Схема созданного в рамках данной магистерской диссертации программного средства представлена на Рисунке 5; этапы, реализованные нами, представлены на ней прямоугольниками и параллелограммами.

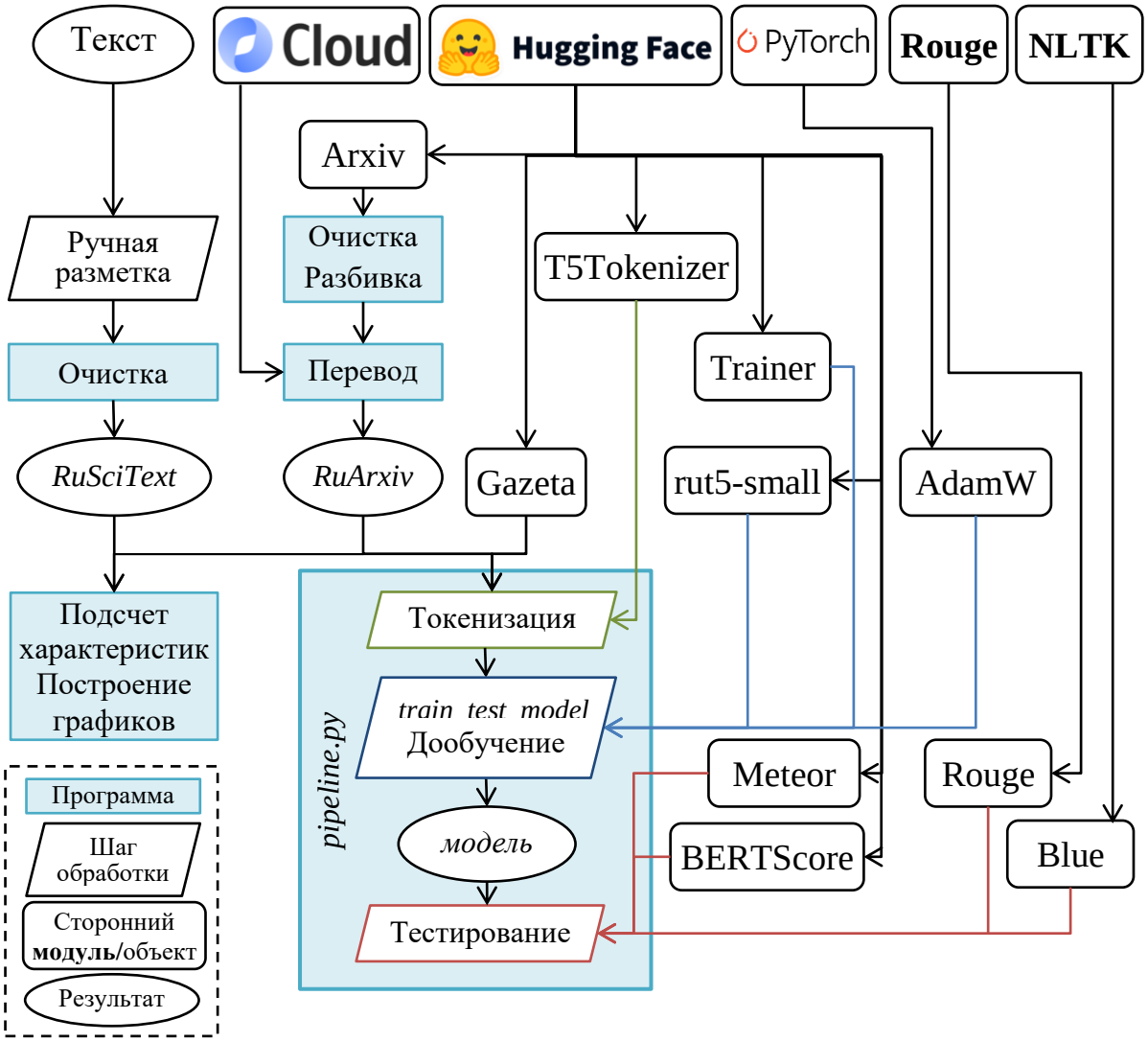



Рисунок 5: Схема реализованного программного средства

## 5.1. Программное средство для дообучения НЯМ

Для дообучения нейронных языковых моделей решению задачи автоматического реферирования была написана программа на языке *Python 3* с использованием средств Hugging Face; она доступна в двух форматах: файла с расширением .py (*pipeline.py*) и файла Jupyter Notebook (*Pipeline.ipynb*). Программа реализует обучение языковой модели и проводит ее тестирование по четырем метрикам: ROUGE, BLEU, METEOR и BERTScore. При необходимости цикл обучения может повторяться.

Программа предоставляет возможность обучения модели с валидацией либо без нее, а также с промежуточным тестированием либо без него. Обучение длится несколько шагов, на каждом из которых модель обучается указанное количество эпох, логируется, затем тестируется. Количество шагов и эпох необходимо передать в функцию *train_test_model* в качестве параметров *num_steps* и *num_epochs* соответственно. При обучении сохраняется не только состояние модели, но и состояние ее оптимизатора, а также значение ошибки (*loss*) на обучающей и валидационной выборках (последнее – если обучение с валидацией). Также можно получить результаты работы дообученной модели на своей тестовой выборке, например, для экспертной оценки.

Стоит также отметить, что полученную программу можно использовать для обучения и других нейронных языковых моделей, представленных с помощью средств платформы Hugging Face.

## 5.2. Дообучение модели rut5-small

С помощью реализованной программы модель rut5-small (ее схема представлена на Рисунке 6) была дообучена на выбранных наборах данных. Использовался оптимизатор *AdamW*[14] со значением *learning rate* = 1e-4.

При дообучении rut5-small необходимо учесть, что модель не может обрабатывать тексты длиннее 20 100 токенов. Также в ходе работы выяснилось, что из-за ограничения в ресурсах мы не сможем обучать модель на слишком больших входах – более 13 000 токенов. По этой причине датасет RuArxiv пришлось сократить. Мы отобрали пары, в которых длина текста не превышает 12 000 токенов, а длина реферата не превышает 1 000 токенов. Также мы поставили условие на то, чтобы процент сжатия был не более 50%, поскольку пары, имеющее больший процент, скорее всего являются ошибочными. Таким образом получился набор данных из 268 пар.

[14] Оптимизатор AdamW: https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html

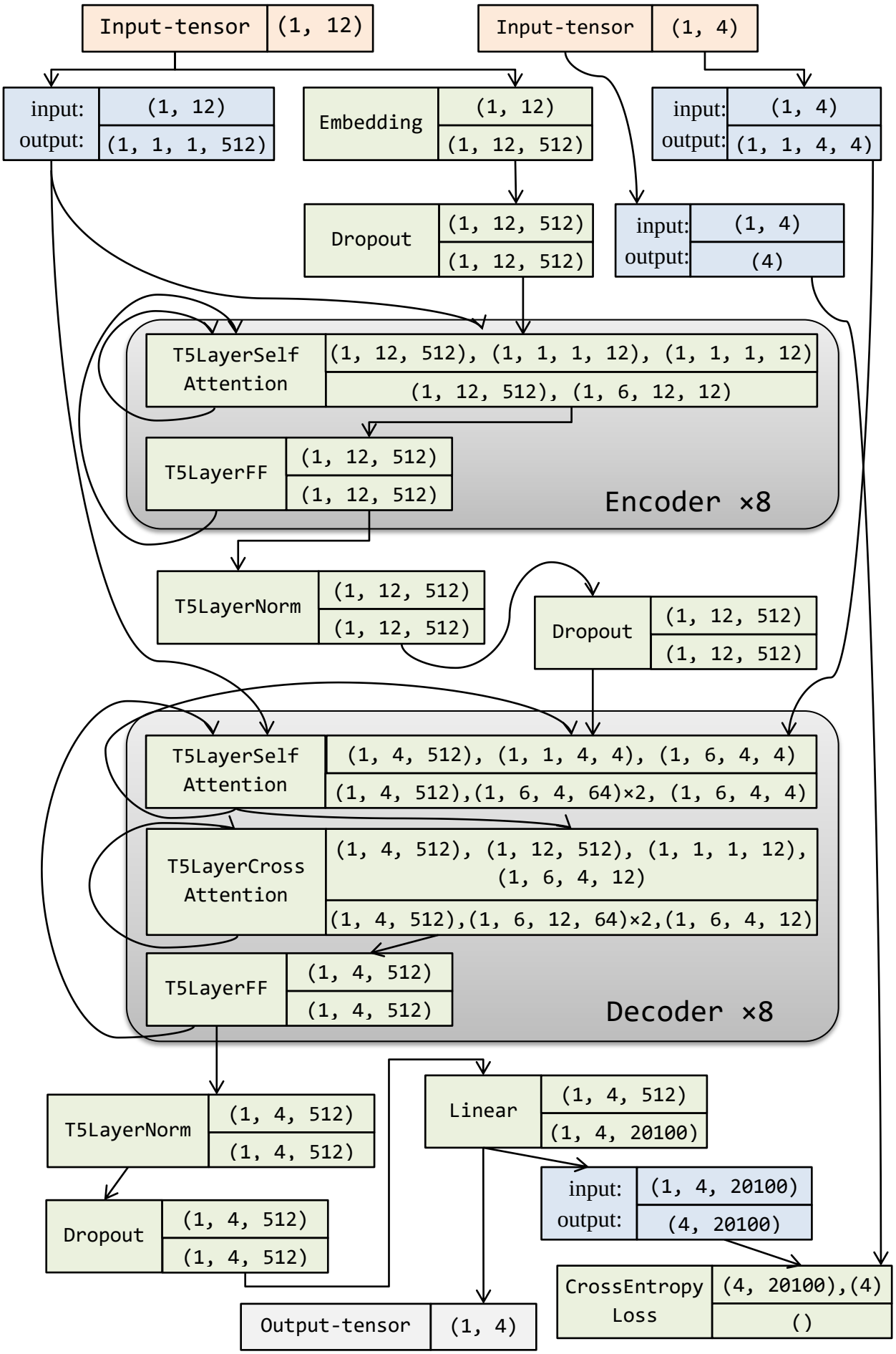



Рисунок 6: Схема модели rut5-small

Датасеты были разделены на обучающую и валидационную выборки в соотношении 9:1 (для RuArxiv – это 241 и 27 пар, для RuSciText – это 90 и 10 пар соответственно). Обучение на каждом наборе длилось, пока не наступало переобучение. В итоге было получено 5 программных моделей. Схема процесса дообучения представлена на Рисунке 7; результаты дообучения представлены на Графиках 1-5.

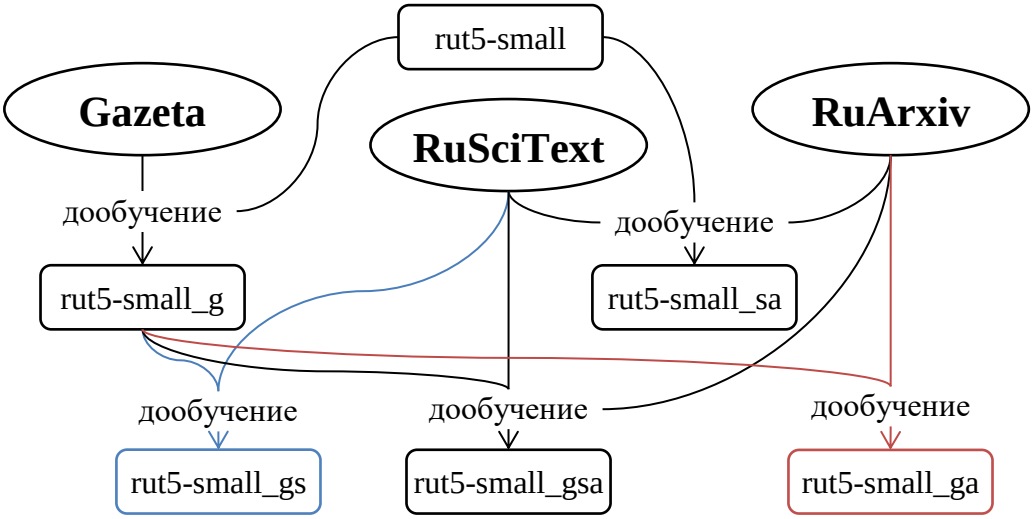



Рисунок 7: Схема дообучения модели rut5-small

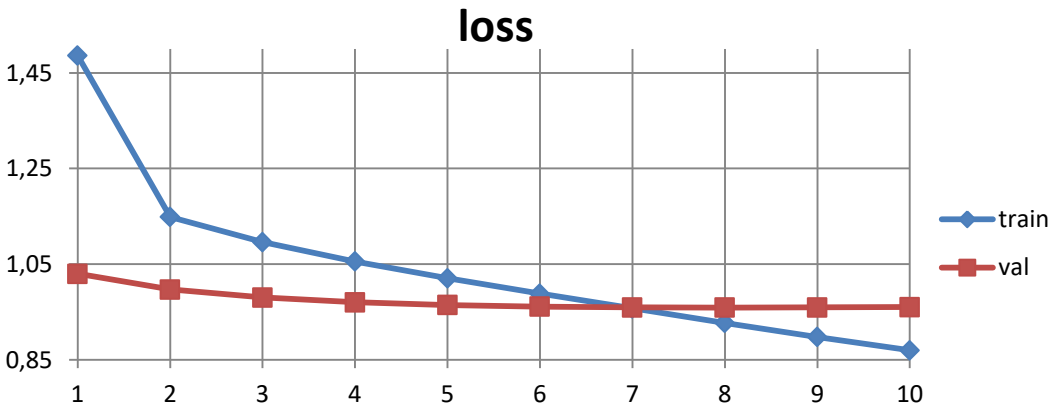



График 1: Обучение модели rut5-small на 10-ти эпохах на Gazeta

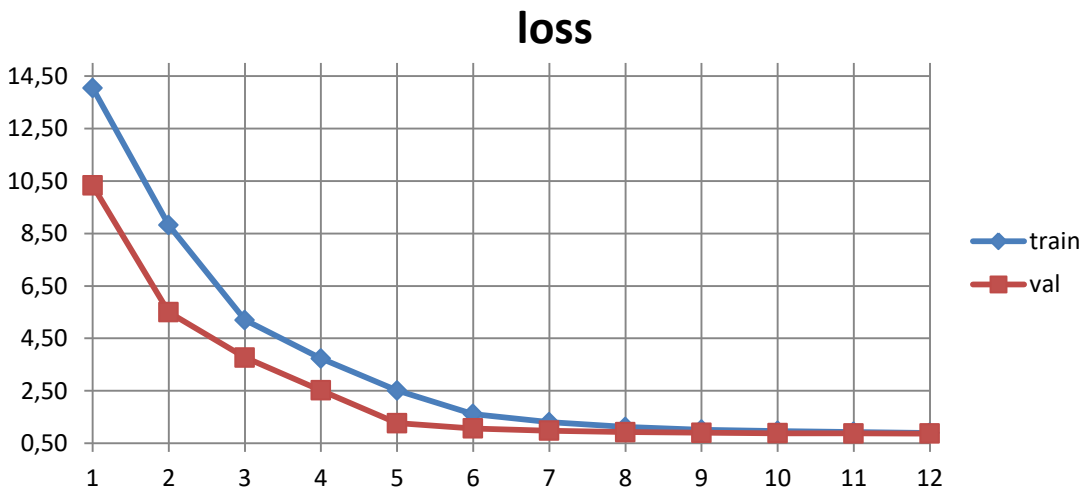



График 2: Обучение модели rut5-small на 12-ти эпохах на RuSciText и RuArxiv

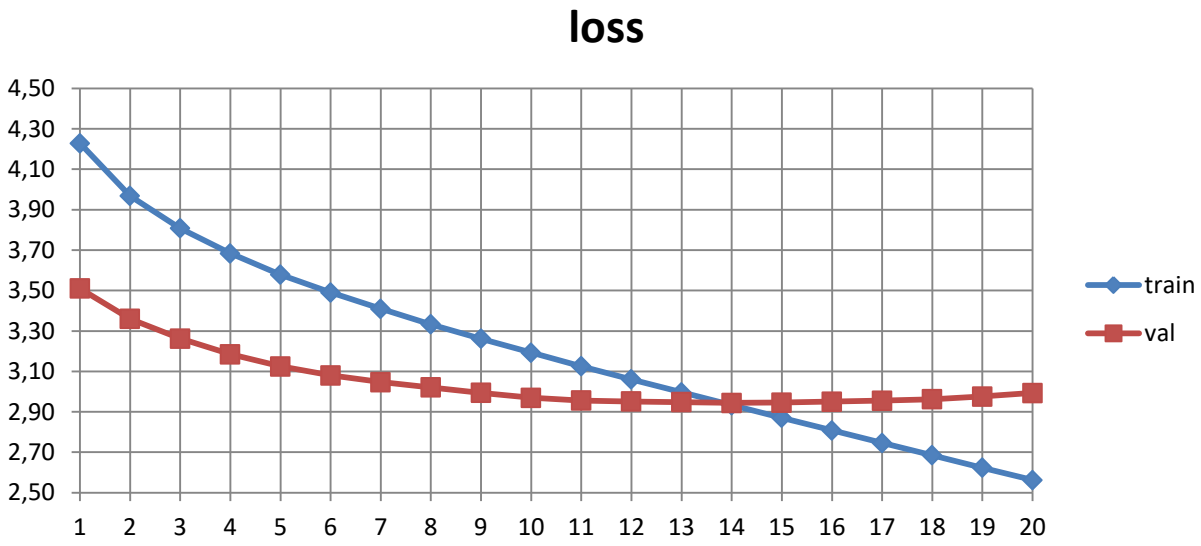



График 3: Обучение модели rut5-small_g на 20-ти эпохах на RuSciText

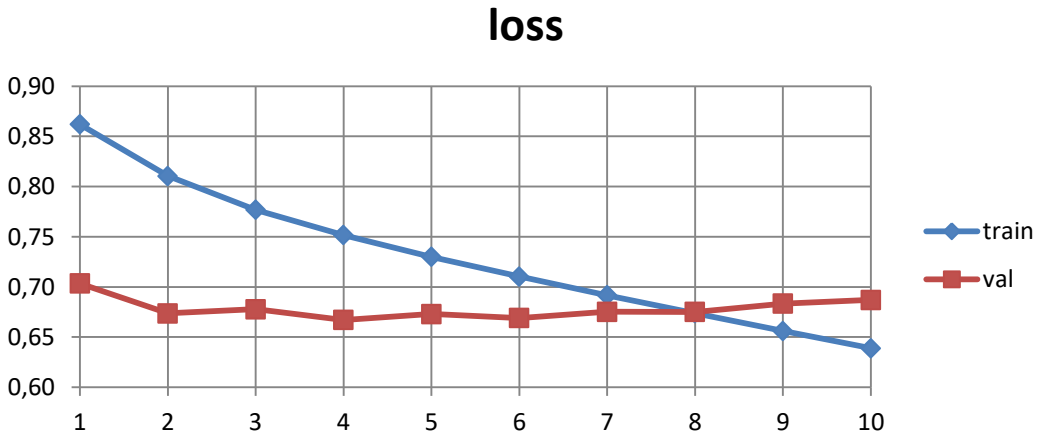



График 4: Обучение модели rut5-small_g на 10-ти эпохах на RuArxiv

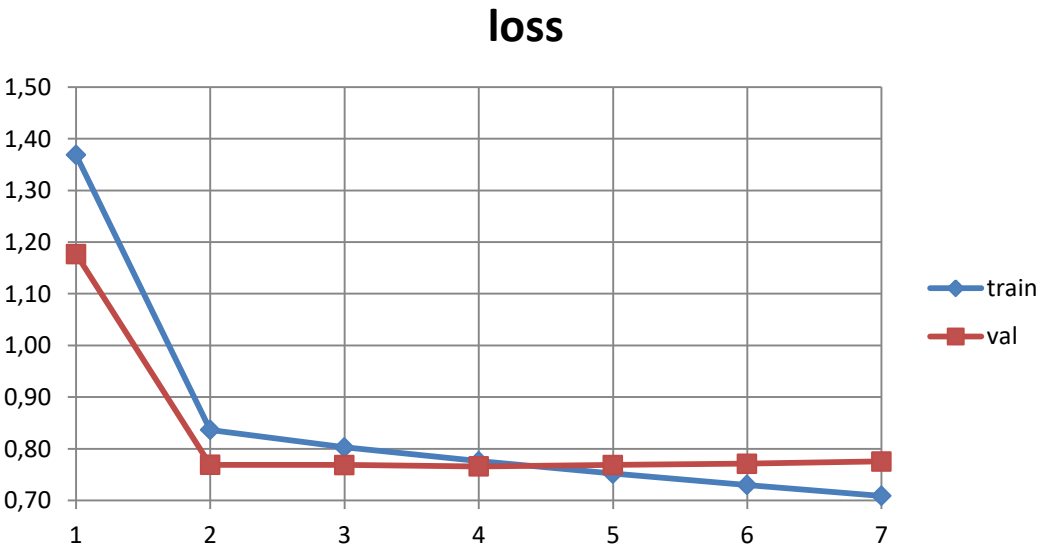



График 5: Обучение модели rut5-small_g на 7-ми эпохах на RuSciText и RuArxiv

На Графике 1 показаны результаты обучения модели rut5-small на датасете Gazeta. Для дальнейших исследований была отобрана модель после 7 эпох обучения. Будем называть ее **rut5-small_g**.

На Графике 2 представлены результаты обучение модели rut5-small на собранных нами наборах данных. Для дальнейших исследований была взята модель после 12 эпох обучения. Назовем полученную модель **rut5-small_sa**.

Для следующих трех моделей дообучение проходило в 2 этапа. Модель rut5-small_g дополнительно дообучалась на датасете RuSciText, результаты показаны на Графике 3. По итогу была взята модель после 14 эпох обучения. Назовем эту модель **rut5-small_gs**.

Также модель rut5-small_g была дообучена на датасете RuArxiv, результаты представлены на Графике 4. Назовем модель, полученную на 8 эпохе обучения, **rut5-small_ga**.

Результаты дообучения модели rut5-small_g на наборах данных RuSciText и RuArxiv представлены на Графике 5. Модель, отобранную на 4 эпохе обучения, будем называть **rut5-small_gsa**.

# 6. Экспериментальное исследование работы моделей семейства T5

## 6.1. Экспертная оценка работы моделей

При анализе работы всех исследуемых моделей кроме значения метрик мы экспертно оценивали:

- качество генерируемого текста (правильное построение предложений, связность);
- покрытие рефератом основных *аспектов* статьи (задача, методы, результаты);
- наличие фактических ошибок.

Данные характеристики сложно оценить автоматически. Современные программные меры для оценки подобных характеристик требуют специальных предобученных нейронных сетей, которых для русского языка на данный момент нет [14, 21].

Работа моделей оценивалась на длинных входных текстах. В качестве демонстрационного примера был взят текст ВКР длинной 5 586 токенов (посчитано с помощью токенизатора *razdel*). Также изучались результаты работы моделей на более коротких текстах тезисов к ВКР [2] и текстах средней длины – 10 текстах валидационной выборки RuSciText. В Приложении С представлены примеры работы всех рассмотренных моделей семейства T5.

### 6.1.1. Анализ работы дообученных моделей

Результаты экспертной оценки показали, что модели, полученные на основе rut5-small, способны генерировать качественные тексты. Однако им не всегда удается выделить ключевые моменты статьи и сгенерировать качественный текст. Также модели могут искажать информацию, например сгенерировать вместо фразы «*менее энергоемкий*» фразу «*более энергоемкий*», использовав слова, встречающиеся в одном контексте, но противоположные по смыслу. Модели также страдают от **галлюцинаций** – частой проблемы нейронных сетей архитектуры Transformer. Для рассматриваемой задачи это выражается в повторении отдельных частей текста или в добавлении информации, не входившей в исходный текст. Например, во фразе «*В работе С.В. Панов назвал цель…*» содержится фамилия и инициалы человека, не являющегося автором статьи.

Итоговые рефераты **rut5-small_g** для коротких текстов чаще всего содержат ключевые термины и информацию о примененных методах, т.е. модель обращает внимание на некоторые индикаторные фразы, показывающие, что в тексте речь идет о методах. Рефераты состоят из 1-2 законченных предложений и всегда заканчиваются

точкой. Для средних и длинных текстов модель чаще галлюцинирует и хуже выделяет важную информацию.

Все остальные модели, дообученные на научных статьях, генерируют тексты в научном стиле. Полученные рефераты длиннее и информативнее, что подтвердило наши предположения о необходимости финального дообучения модели на более длинных эталонных рефератах. Однако стоит отметить, что собранные нами наборы данных недостаточно большие, особенно RuSciText. Несмотря на полученные графики обучения (см. Графики 2-3) при анализе результатов создается впечатление, что эти модели переобучились. **rut5-small_sa** и **rut5-small_gs** чаще остальных страдают от галлюцинаций, причем модели повторяют не только фразы, но и отдельные буквы. Для этих моделей возникает необходимость дополнительно ограничивать длину выхода, т.е. указывать параметр *max_length* равным 100-200, для того, чтобы избежать галлюцинаций, которыми модель «заполняет» оставшееся количество токенов.

Модель **rut5-small_sa** часто не может закончить одно предложение, создавая большие сложноподчиненные конструкции. В противоположность ей модель **rut5-small_ga** использует простые предложения:

«*… Это было проведено с помощью бинарной классификации для взаимного дополнения и развития. Это было проведено с помощью достаточно простой однослойной нейронной сети для взаимного дополнения и развития. Это было проведено с помощью программного комплекса CVSS. Это позволяет рассчитывать параметры кровотока в любой точке системы сосудов человека, как в норме, так и патологии*».

Модель **rut5-small_gsa** генерирует достаточно длинные и информативные рефераты, состоящие из 2-4 законченных сложных предложения, без нагромождений сложных конструкций и всегда оканчивающиеся точкой. При этом дополнительно ограничивать по длине ее выход не требовалось.

### 6.1.2. Анализ работы моделей размера Base

Напомним, что для анализа нами были выбраны две языковые модели архитектуры T5 размера Base: модель **ruT5-base-absum**, дообученная на задачу реферирования на датасетах Gazeta, XL-Sum, MLSUM и WikiLingua, и модель **rut5_base_sum_gazeta**, дообученная на датасете Gazeta. Модель **rut5_base_sum_gazeta** показала результаты лучше, чем **ruT5-base-absum**, а именно:

- Модель лучше передавала смысл статьи как на длинных, так и на коротких входных данных.
- Сгенерированные рефераты чаще состояли более чем из одного предложения.

- С точки зрения смысла предложения получались законченными и всегда оканчивались точкой в отличие от модели **ruT5-base-absum**.

Такие результаты можно объяснить более длинными эталонными рефератами в обучающей выборке модели **rut5_base_sum_gazeta**, дообученной только на датасете Gazeta с более качественно обработанными текстами.

Модели оказались не склонны выделять какие-либо аспекты исходного текста и хуже отражали важную для научных статей информацию из длинных тестовых примеров. Выходы модели содержали меньше фактических ошибок и генерировали более качественные предложения. Заметим, что обе модели также страдают от галлюцинаций, особенно на длинных текстах.

## 6.2. Сравнение моделей на датасете Gazeta

Для всех семи рассмотренных нами моделей была проведена оценка их работы на тестовой выборке датасете Gazeta. Для подсчета метрик использовалось созданное нами программное средство. Значения полученных метрик представлены в Таблице 6 вместе с результатами из исследования [17].

Таблица 6: Сравнение результатов работы НЯМ на датасете Gazeta

| Модель | ROUGE-1 | ROUGE-2 | ROUGE-L | BLUE | METEOR | BERTScore |
|---|---|---|---|---|---|---|
| mBART | 31,55 | 13,54 | 28,22 | **11,19** | **34,09** | 56,56 |
| ruGPT3Small | 18,84 | 4,06 | 16,68 | 3,13 | 18,70 | 44,06 |
| ruGPT3Large | 23,45 | 6,45 | 20,73 | 4,93 | 23,77 | 47,76 |
| ruT5-large | **32,45** | **13,97** | **29,24** | 10,88 | 31,21 | **57,73** |
| ruT5-base | 30,45 | 12,63 | 27,41 | 9,54 | 28,69 | 56,35 |
| ruT5-base-absum | 16,75 | 5,61 | 15,32 | 1,7 | 12,92 | *70,67* |
| rut5_base_sum_gazeta | 22,43 | 8,2 | 20,57 | 5,77 | 22,92 | ***72,35*** |
| rut5-small_g | 20,96 | 7,13 | 19,35 | 4,75 | 20,83 | ***71,04*** |
| rut5-small_sa | 13,89 | 1,95 | 12,59 | 0,31 | 9,74 | *58,91* |
| rut5-small_gs | 13,31 | 1,7 | 11,9 | 0,93 | 10,31 | *66* |
| rut5-small_ga | 21,69 | 7,33 | 20,12 | 3,15 | 22,9 | *67,53* |
| rut5-small_gsa | 15,56 | 4,44 | 14,56 | 1,53 | 16,87 | *62,46* |

Результаты тестирования показывают, что качество модели T5 размера Small сильно хуже моделей этого же семейства размера Base и Large из исследования [17]. Поэтому начальная гипотеза о том, что дообучение модели небольшого размера на

дополнительных данных позволит приблизить показатели ее качества к показателям других моделей семейства T5, не подтвердилась. При этом модели **rut5-small_g** и **rut5-small_ga** обходят по качеству модель ruGPT3Small и очень близки к показателям модели ruGPT3Large, несмотря на разницу в размерах: ruGPT3Small почти в 2 раза больше, а ruGPT3Large почти в 11 раз больше rut5-small.

Интересным наблюдением является то, что показатели модели **rut5-small_ga**, дообученной на RuArxiv, в среднем выше, чем у модели **rut5-small_g**. При этом значение метрики BERTScore, оценивающей схожесть эмбендингов сгенерированного и эталонного текстов, ниже. Возможно, такие результаты обусловлены тем, что **rut5-small_ga** генерирует более длинные тексты, что приводит к большему покрытию n-грамм эталонного реферата.

Значения метрик остальных программных моделей значительно хуже. Модель **rut5-small_sa**, никогда не обучавшаяся на датасете Gazeta, ожидаемо показывает худшие результаты по всем метрикам, кроме ROUGE.

Модели **ruT5-base-absum** и **rut5_base_sum_gazeta** демонстрируют результаты хуже, чем модель размера Base из исследования [17], несмотря на большее количество параметров. Результаты генерации этих моделей на тестовой выборке датасета Gazeta выложены на GitHub вместе с проведенными экспериментами.

Дополнительно отметим, что результаты BERTScore для всех рассматриваемых нами моделей получились сильно выше, чем у всех остальных моделей, что выглядит довольно странно. Код с расчетом метрик также предоставлен вместе со всеми результатами экспериментов на GitHub, автор будет рад получить обратную связь по этому вопросу.

# 7. Заключение

В ходе выполнения магистерской диссертации получены следующие результаты:

1. Проведен анализ современных методов абстрактного реферирования и по его результатам для исследования выбраны модели семейства T5.
2. Проведено исследование русскоязычных наборов данных и собраны собственные наборы данных: размеченный вручную RuSciText и переведенный с английского RuArxiv. Для всех наборов проведено статистическое исследование их характеристик.
3. Создано программное средство для дообучения языковых моделей для решения задачи автоматического реферирования. С его помощью проведено дообучение модели rut5-small.
4. Проведено экспериментальное исследование моделей семейства T5, полученные результаты проанализированы.

Анализ результатов проведенных экспериментов показывает, что дообучение языковых моделей на данных с более длинными эталонными рефератами приводит к генерации более длинных и информативных текстов. Дообученные модели rut5-small_g и rut5-small_ga, несмотря на их размер, демонстрируют сравнимые с другими моделями результаты. Модели большего размера генерируют более связный текст и допускают меньше фактических ошибок. Также все модели склонны к галлюцинациям при обработке длинных текстов.

Результаты экспериментов, код разработанного программного средства, а также ссылки на архив обученных по эпохам моделей и их оптимизаторов выложены на GitHub: https://github.com/Svetych/RuSciTextSum.

Обработанная текстовая коллекция и полученный из нее набор данных RuSciText выложены на диск: https://disk.yandex.ru/d/UW1_0JASBxJUbw.

Текста, получившиеся в ходе формирования набора данных RuArxiv, и итоговый набор выложены на диск: https://disk.yandex.ru/d/c_gE8O4fpPj1_w.

Файл Jupyter Notebook с подсчитанными статистическими характеристиками датасетов выложен на диск: https://drive.google.com/file/d/11QTGgnUD2cue03Dd7Ba4DVtcR6gwL2Ho/view?usp=sharing.

## 8. Список литературы


1. ГОСТ "ГОСТ 7.9-95 (ИСО 214-76) СИБИД" от 26.04.95 № 7.9-95 // Комитет Российской Федерации по стандартизации, метрологии и сертификации. – 27.02.96 г. – № 108.
2. Тезисы 2020 / [Электронный ресурс] // Сборник тезисов лучших выпускных квалификационных работ факультета ВМК МГУ 2020 года : [сайт]. – URL: https://cs.msu.ru/sites/cmc/files/attachs/sbornik_vkr_vmk_2020.pdf?ysclid=l2c5y0hwi3 (дата обращения: 07.05.2024).
3. Aggarwal C. C. Machine learning for text. - Second Edition - Switzerland AG: Springer Nature, 2022.
4. Ahuir V. et al. Abstractive Summarizers Become Emotional on News Summarization // Applied Sciences. – 2024. – Т. 14. – №. 2. – С. 713.
5. Aswani S. et al. Automatic text summarization of scientific articles using transformers – A brief review // Journal of Autonomous Intelligence. – 2024. – Т. 7. – №. 5.
6. Banerjee S., Lavie A. Meteor: an automatic metric for MT evaluation with high levels of correlation with human judgments // Proceedings of ACL-WMT. – 2007. – С. 228–231.
7. Bhat I. K., Mohd M., Hashmy R. SumItUp: A hybrid single-document text summarizer // Soft Computing: Theories and Applications: Proceedings of SoCTA 2016, Volume 1. – Springer Singapore, 2018. – С. 619-634.
8. Cohan A. et al. A discourse-aware attention model for abstractive summarization of long documents // Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. - New Orleans, Louisiana: Association for Computational Linguistics, 2018. - С. 615-621.
9. Dataset BOOKSUM: A Collection of Datasets for Long-form Narrative Summarization / [Электронный ресурс] // GitHub : [сайт]. – URL: https://github.com/salesforce/booksum (дата обращения: 07.05.2024).
10. Dataset Card for Sci Lay / [Электронный ресурс] // Hugging Face : [сайт]. – URL: https://huggingface.co/datasets/paniniDot/sci_lay (дата обращения: 07.05.2024).
11. Deutsch D., Dror R., Roth D. Re-examining system-level correlations of automatic summarization evaluation metrics / Deutsch D., Dror R., Roth D. [Текст] // Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. – Seattle, United States: Association for Computational Linguistics, 2022. – С. 6038-6052.
12. Devlin J. et al. BERT: Pre-training of deep bidirectional transformers for language understanding // arXiv preprint arXiv:1810.04805. – 2018.

13. El-Kassas W. S. et al. Automatic text summarization: A comprehensive survey // Expert systems with applications. – 2021. – Т. 165. – С. 113679.
14. Gao M. et al. Human-like summarization evaluation with ChatGPT // arXiv preprint arXiv:2304.02554. – 2023.
15. Gazeta dataset / [Электронный ресурс] // GitHub : [сайт]. – URL: https://github.com/IlyaGusev/gazeta (дата обращения: 07.05.2024).
16. Gusev I. Dataset for automatic summarization of Russian news // Artificial Intelligence and Natural Language: 9th Conference, AINL 2020, Helsinki, Finland, October 7–9, 2020, Proceedings 9. – Springer International Publishing, 2020. – С. 122-134.
17. Goloviznina V., Kotelnikov E. Automatic summarization of Russian texts: Comparison of extractive and abstractive methods // Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference “Dialogue 2022”. – М.: 2022.
18. Hasan T. et al. XL-sum: Large-scale multilingual abstractive summarization for 44 languages // Findings of the Association for Computational Linguistics: ACL-IJCNLP. – Association for Computational Linguistics, 2021. – С. 4693-4703.
19. Hovy E., Lin C. Y. Automated text summarization and the SUMMARIST system // TIPSTER TEXT PROGRAM PHASE III: Proceedings of a Workshop held at Baltimore, Maryland. - Association for Computational Linguistics, USA, October 13-15, 1998. – 1998. – С. 197-214.
20. Hugging Face The AI community / [Электронный ресурс] // Hugging Face : [сайт]. – URL: https://huggingface.co/ (дата обращения: 07.05.2024).
21. Koto F., Baldwin T., Lau J. H. Ffci: A framework for interpretable automatic evaluation of summarization // Journal of Artificial Intelligence Research. – 2022. – Т. 73. – С. 1553–1607.
22. Kryściński W. et al. BookSum: A collection of datasets for long-form narrative summarization // arXiv preprint arXiv:2105.08209. – 2021.
23. Ladhak F. et al. WikiLingua: A New Benchmark Dataset for Cross-Lingual Abstractive Summarization // Findings of the Association for Computational Linguistics: EMNLP 2020. - Association for Computational Linguistics, 2020. - С. 4034-4048.
24. Li J. et al. Pre-trained Language Models for Text Generation: A Survey // ACM Computing Surveys. – 2021.
25. Lin C. Y. ROUGE: A package for automatic evaluation of summaries // Text summarization branches out. – Barcelona, Spain. – 2004. – С. 74-81.
26. Nallapati R. et al. Abstractive text summarization using sequence-to-sequence RNNs and beyond // Proceedings of the 20th Conference on Computational Natural Language

Learning. – Berlin, Germany: Association for Computational Linguistics, 2016. – С. 280-290.

27. Papineni K. et al. BLEU: a Method for Automatic Evaluation of Machine Translation // Proceedings of the 40th annual meeting of the Association for Computational Linguistics. – Philadelphia. – 2002. – С. 311-318.
28. Radford A. et al. Improving language understanding by generative pre-training. – 2018.
29. Raffel C. et al. Exploring the limits of transfer learning with a unified text-to-text transformer // Journal of machine learning research. – 2020. – Т. 21. – №. 140. – С. 1-67.
30. Phang J., Zhao Y., Liu P. J. Investigating efficiently extending transformers for long input summarization // arXiv preprint arXiv:2208.04347. – 2022.
31. Ramya R. S. et al. A Survey on Automatic Text Summarization and its Techniques // International Journal of Intelligent Systems and Applications in Engineering. – 2023. – Т. 11. – №. 1s. – С. 63-71.
32. Razdel – сегментация русскоязычного текста на токены и предложения / [Электронный ресурс] // razdel : [сайт]. – URL: https://natasha.github.io/razdel/ (дата обращения: 07.05.2024).
33. Scialom T. et al. MLSUM: The multilingual summarization corpus // arXiv preprint arXiv:2004.14900. – 2020.
34. Tejaswin P., Naik D., Liu P. How well do you know your summarization datasets? // Findings of the Association for Computational Linguistics. – Association for Computational Linguistics, 2021. - С. 3436-3449.
35. Vaswani A. et al. Attention is all you need // Advances in neural information processing systems 30. – 2017. – Т. 30.
36. WikiLingua: A Multilingual Abstractive Summarization Dataset / [Электронный ресурс] // GitHub : [сайт]. – URL: https://github.com/esdurmus/Wikilingua (дата обращения: 07.05.2024).
37. WordNet A Lexical Database for English / [Электронный ресурс] // WordNet : [сайт]. – URL: https://wordnet.princeton.edu/ (дата обращения: 07.05.2024).
38. XL-Sum / [Электронный ресурс] // GitHub : [сайт]. – URL: https://github.com/csebuetnlp/xl-sum (дата обращения: 07.05.2024).
39. Yadav D., Desai J. and Yadav A. K. Automatic Text Summarization Methods: A Comprehensive Review // arXiv preprint arXiv: arXiv:2204.01849. – 2022.
40. Zhang J. et al. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization // 37 th International conference on machine learning. – Vienna, Austria: PMLR, 2020. – С. 11328-11339.

41. Zhang M. et al. ROUGE-SEM: Better evaluation of summarization using ROUGE combined with semantics // Expert Systems with Applications. – 2024. – T. 237. – C. 121364.
42. Zhang T. et al. BERTScore: Evaluating Text Generation with BERT // arXiv preprint arXiv:1904.09675. – 2019.

# Приложение А

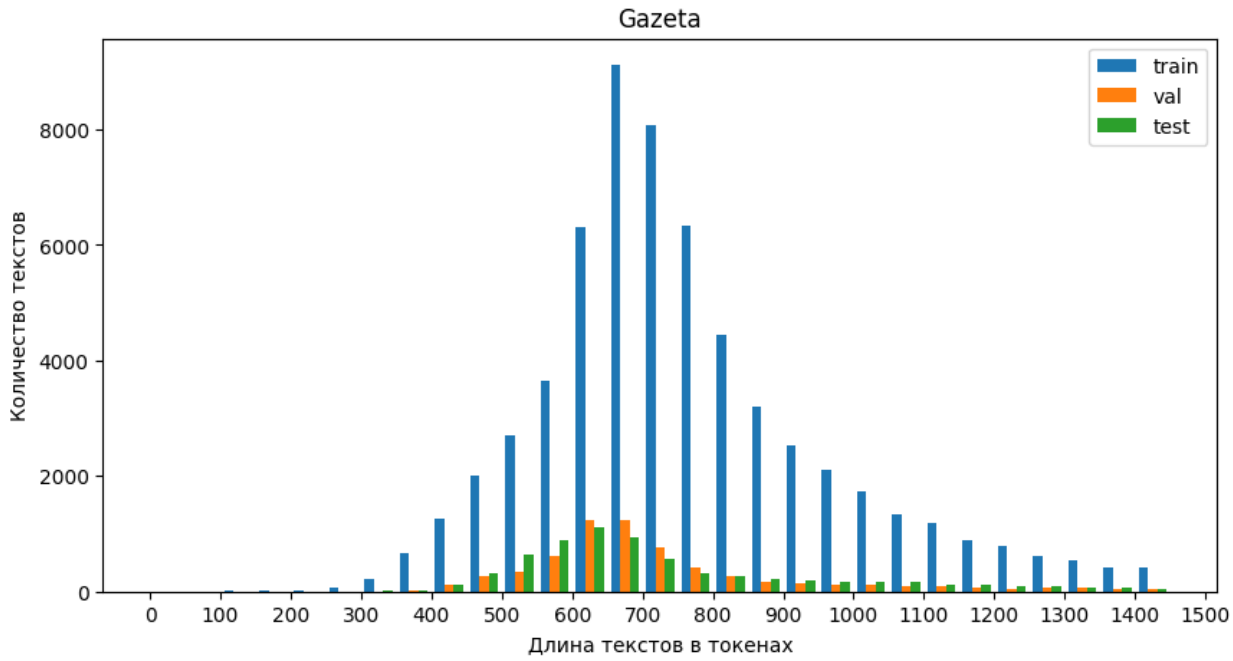
## Характеристики русскоязычных наборов данных

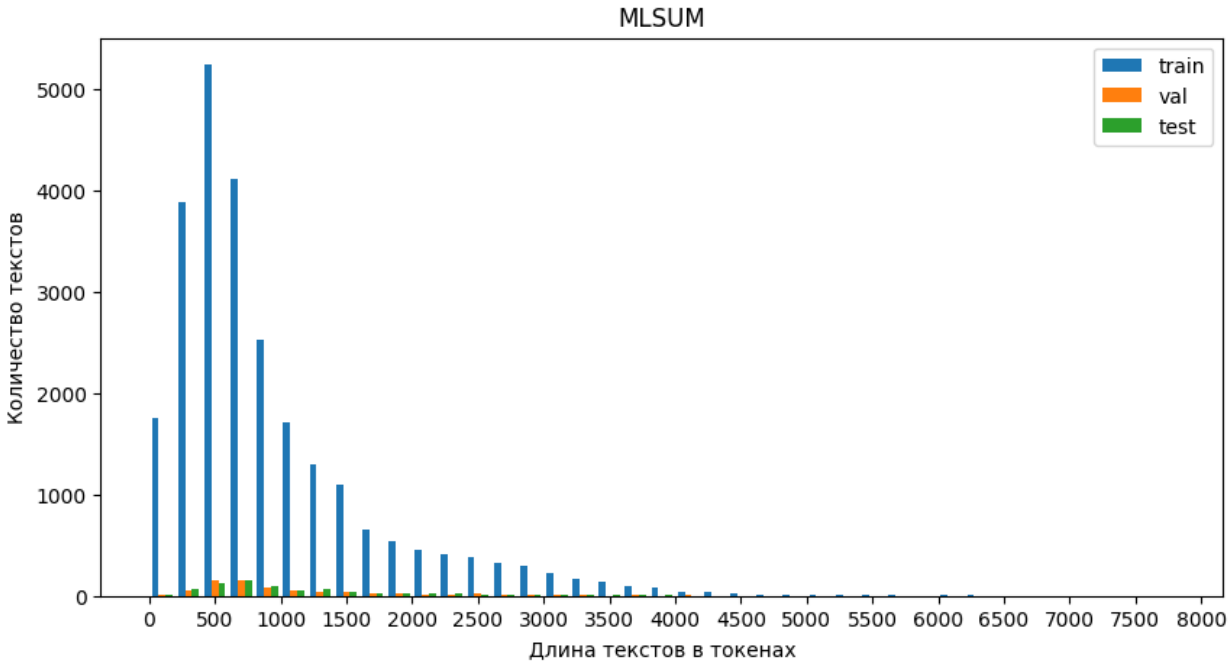
Полученные результаты исследования русскоязычных наборов данных:

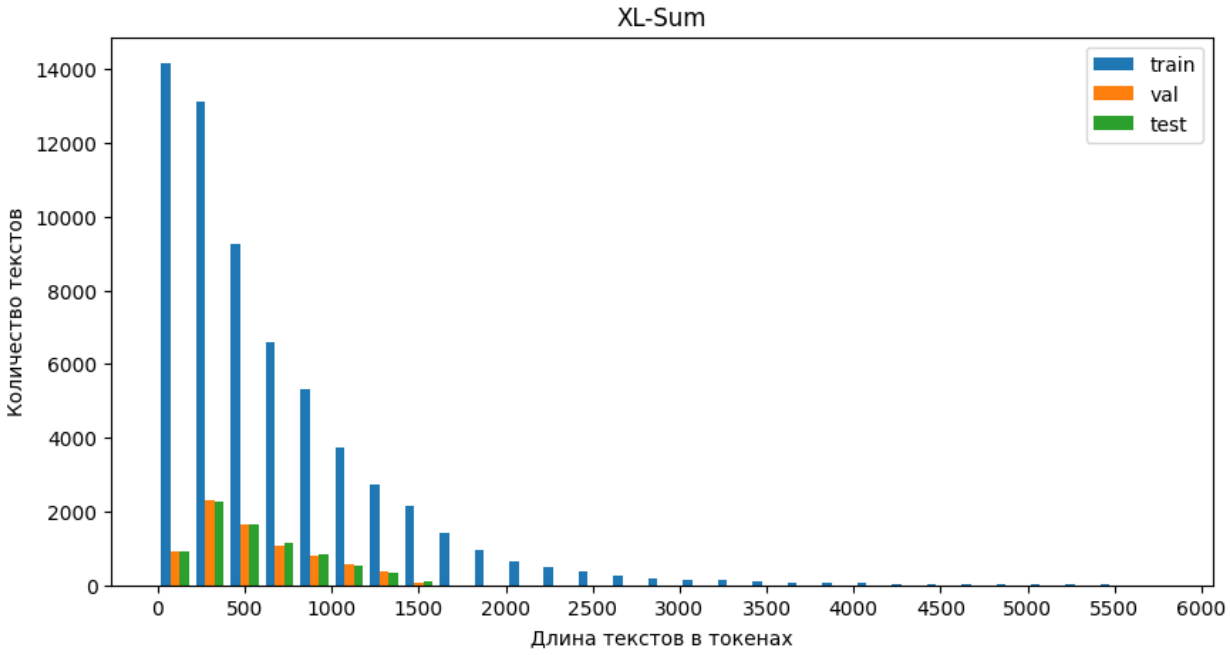
| Датасет | Выборки | Размер | Данные | Длина в токенах | | |
|---|---|---|---|---|---|---|
| | | | | min | max | mean |
| Gazeta | train | 60 964 (82.2%) | текст | 28 | 1 500 | 766,88 |
| | | | реферат | 15 | 85 | **49,57** |
| | | | *сжатие* | *1,11 %* | *178,57 %* | ***6,83 %*** |
| | validation | 6 369 (8.6%) | текст | 344 | 1 500 | 723,95 |
| | | | реферат | 15 | 85 | 53,09 |
| | | | *сжатие* | *1,15 %* | *19,6 %* | *7.72 %* |
| | test | 6 793 (9.2%) | текст | 246 | 1 500 | 732,04 |
| | | | реферат | 15 | 85 | 53,91 |
| | | | *сжатие* | *1,2 %* | *23,68 %* | *7,83 %* |
| MLSUM | train | 25 556 (94.4%) | текст | 55 | 11 689 | 949,86 |
| | | | реферат | 10 | 65 | **14,66** |
| | | | *сжатие* | *0,09 %* | *50,0 %* | ***2,97 %*** |
| | validation | 750 (2.8%) | текст | 118 | 5 842 | 1 156,67 |
| | | | реферат | 10 | 30 | 13,43 |
| | | | *сжатие* | *0,22 %* | *11,11 %* | *1,82 %* |
| | test | 757 (2.8%) | текст | 69 | 26 794 | 1 214,4 |
| | | | реферат | 10 | 35 | 13,44 |
| | | | *сжатие* | *0.07 %* | *27,54 %* | *1,88 %* |
| XL-Sum | train | 62 243 (80%) | текст | 19 | 22 274 | 682,14 |
| | | | реферат | 1 | 246 | **29,4** |
| | | | *сжатие* | *0,01 %* | *75,0 %* | ***9,56 %*** |
| | validation | 7 780 (10%) | текст | 62 | 1 583 | 556,87 |
| | | | реферат | 8 | 60 | 27,94 |
| | | | *сжатие* | *0.66 %* | *22.58 %* | *6,92 %* |
| | test | 7 780 (10%) | текст | 54 | 1 745 | 555,8 |
| | | | реферат | 8 | 60 | 27,9 |
| | | | *сжатие* | *0,78 %* | *27,78 %* | *6,92 %* |

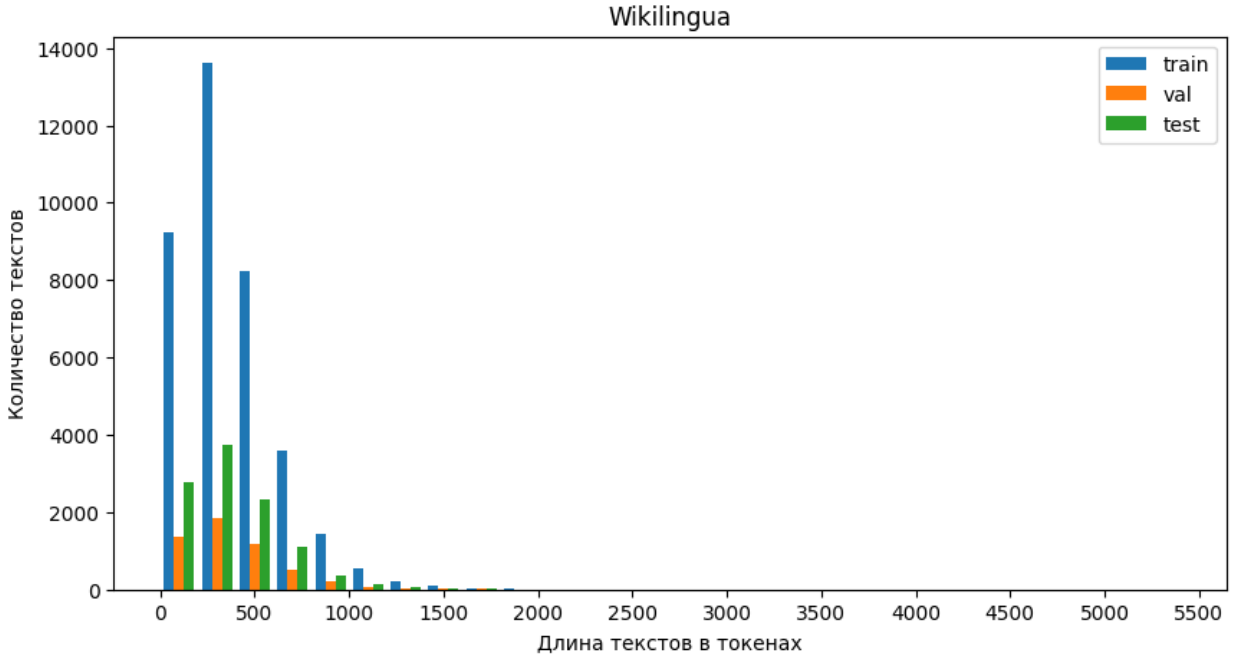
| | | | | | | |
|---|---|---|---|---|---|---|
| WikiLingua | train | 37 029 (70%) | текст | 1 | 4 216 | 378,3 |
| | | | реферат | 2 | 917 | **38,3** |
| | | | *сжатие* | *0,4 %* | *14 200 %* | ***20,72 %*** |
| | validation | 5 289 (10%) | текст | 3 | 4 697 | 383,1 |
| | | | реферат | 3 | 517 | 39,9 |
| | | | *сжатие* | *0,17 %* | *1 533,33 %* | *23,31 %* |
| | test | 10 581 (20%) | текст | 4 | 5 354 | 373,7 |
| | | | реферат | 2 | 464 | 39,2 |
| | | | *сжатие* | *0,28 %* | *2 250 %* | *23,84 %* |

Графики, полученные нами в ходе анализа распределения текстов датасетов по их длинам (в токенах):

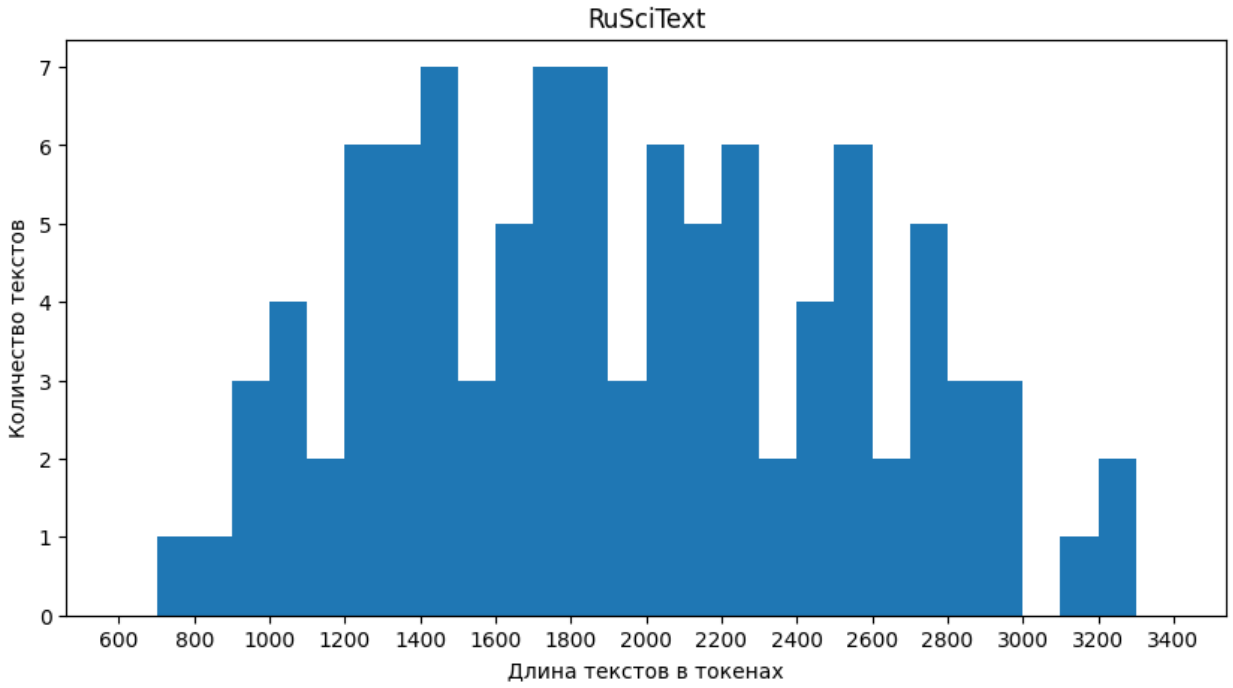


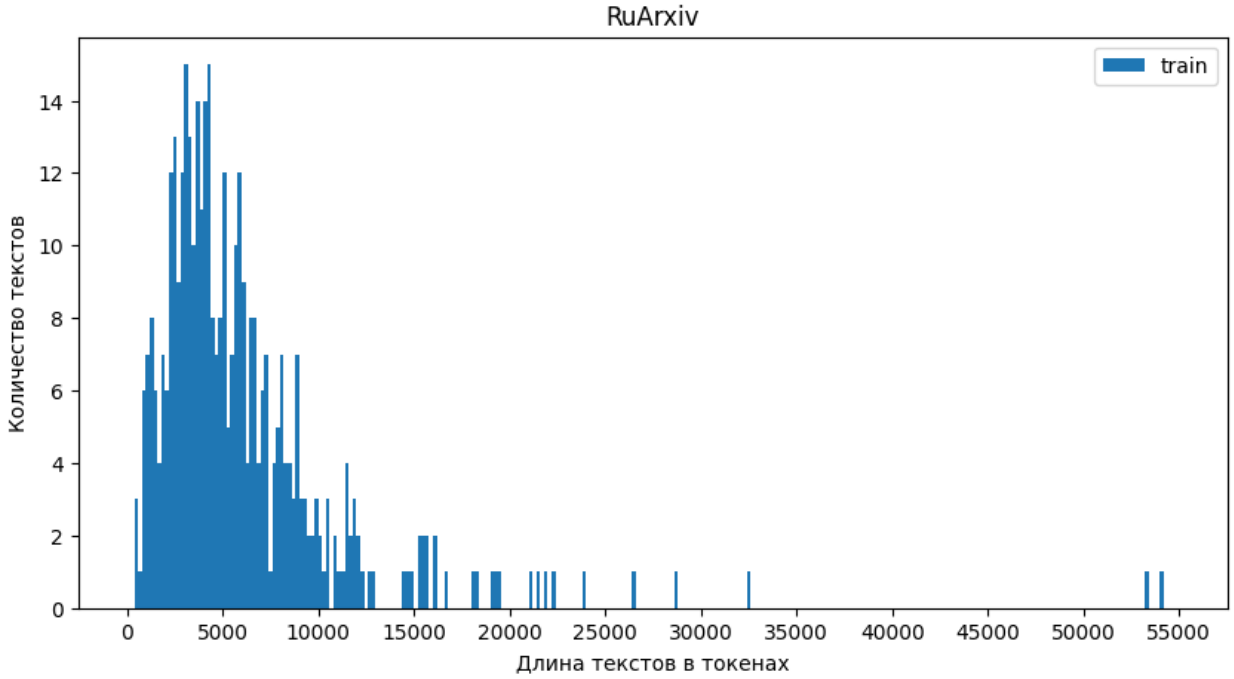

MLSUM
train
val
test
Количество текстов
Длина текстов в токенах


XL-Sum
train
val
test
Количество текстов
Длина текстов в токенах


Wikilingua
train
val
test
Количество текстов
Длина текстов в токенах

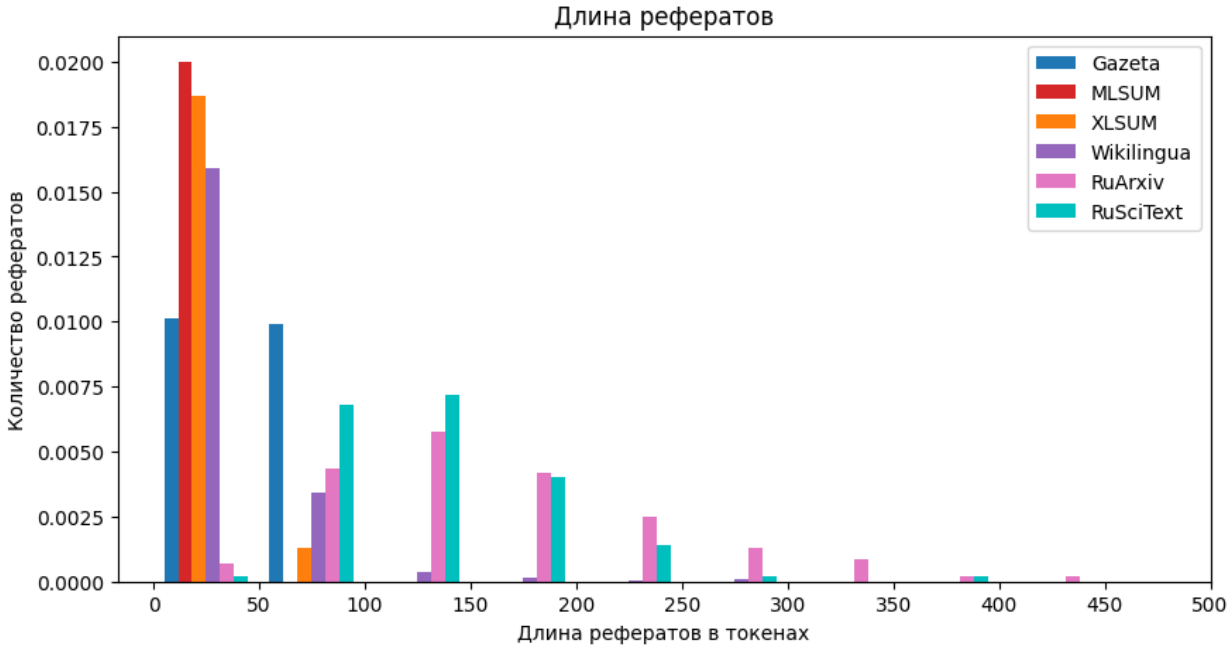
Графики распределения текстов в собранных нами наборах данных по длинам (в токенах):



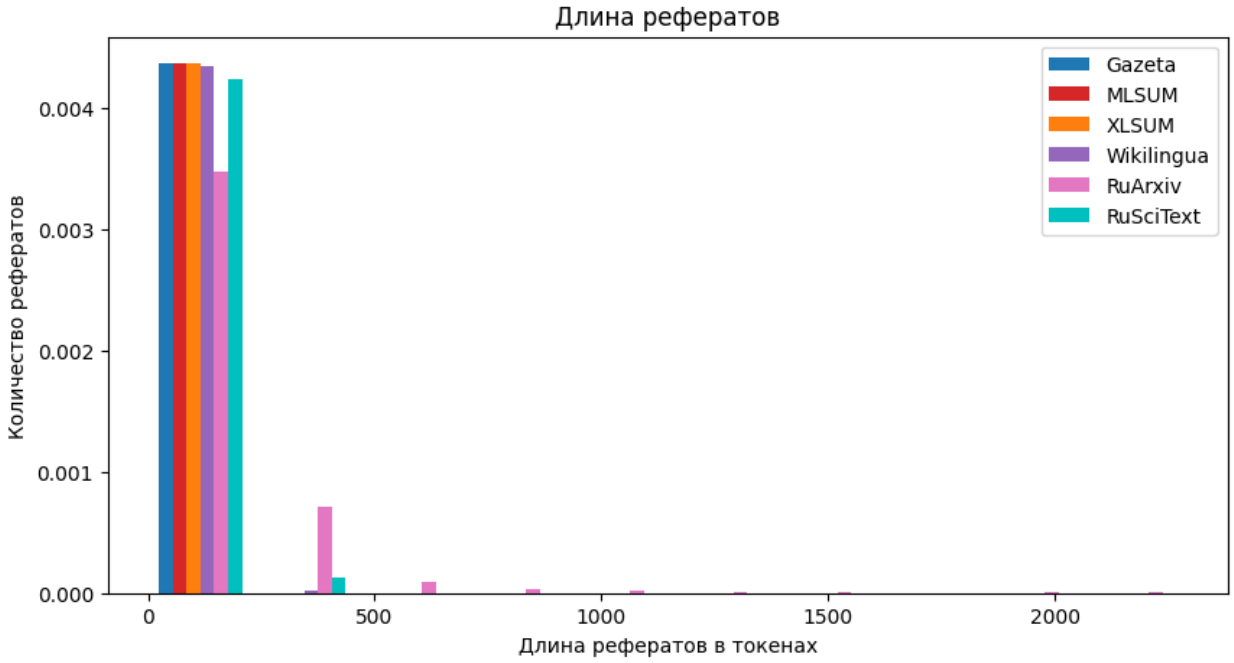



Нормализованные графики распределения эталонных рефератов из всех рассмотренных наборов данных (в токенах).

**С отсечением длины в 500 токенов**:




**Без отсечения**:



# Приложение В

## Метрики для оценки качества автоматического реферирования

**ROUGE** (Recall-Oriented Understudy for Gisting Evaluation) [25] – набор метрик и соответствующее программное средство. Основная идея заключается в подсчете количества *перекрываемых* условных единиц, т.е. n-грамм, которые встречаются и в сгенерированном, и в эталонном реферате одновременно. Так, ROUGE-1 считает отношение к перекрытию отдельных слов (униграм), ROUGE-2 – биграм, а ROUGE-L – n-граммы самой длинной совпадающей подпоследовательности. Точность ($ROUGE_{precision}$) считается как отношение перекрываемых n-грамм к количеству n-грамм в сгенерированном реферате, полнота ($ROUGE_{recall}$) – в эталонном реферате, и $f_n$-мера ($F_N$-ROUGE) – среднее гармоническое значение между точностью и полнотой с параметром N.

$$\text{ROUGE-N} = \frac{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)},$$

где

n – длина n-грамм;

$Count_{match}(gram_n)$ – количество n-грамм ($gram_n$), встречающихся одновременно и в сгенерированном, и в эталонном реферате;

$Count(gram_n)$ – количество n-грамм (для точности – в сгенерированном реферате, для полноты – в эталонном).

**BLEU** (Bilingual Evaluation Understudy) [27]. Изначальная идея заключается в вычислении соответствия между машинным и человеческим переводом. Для задачи реферирования рассматриваются сгенерированные и эталонные тексты рефератов. Отдельные фрагменты сгенерированного текста оцениваются аналогично $ROUGE_{precision}$. Обычно считается перекрытие уни-, би-, три- и тетраграмы. Затем используется средневзвешенное значение этих оценок [27]:

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{gram_n \in C} Count_{clip}(gram_n)}{\sum_{C \in \{Candidates\}} \sum_{gram_n \in C} Count(gram_n)},$$

$$BP = \begin{cases} 1 \; if \; c > r \\ e^{1-\frac{r}{c}} \; if \; c \leq r \end{cases},$$

$$BLEU = BP \cdot e^{\sum_{n=1}^{N} w_n \log p_n},$$

где

$Count_{clip}(gram_n)$ – количество n-грамм, встречающихся одновременно и в сгенерированном, и в эталонном реферате;

Count($gram_n$) – количество n-грамм в сгенерированном реферате;

|c| - длина сгенерированного реферата;

|r| - длина эталонного реферата;

$w_n = \frac{1}{N}$ ; n = 4.

**METEOR** (Metric for Evaluation of Translation with Explicit ORdering) [6]. В отличие от BLEU вычисляется $F_3$-ROUGE мера вместе с функцией разбиения n-грамм на группы (chunk) и сопоставлением синонимов (в соответствии с WordNet [37]). Порядок слов учитывается благодаря Pen (chunk penalty) – штрафу за неправильный порядок групп.

$$F_{mean} = \frac{P \cdot R}{\alpha \cdot P + (1-\alpha) \cdot R},$$

$$Pen = \gamma \cdot frag^{\beta},$$

$$score = (1 - Pen) \cdot F_{mean},$$

где

P – точность (доля n-грамм в сгенерированном реферате, которые также были в эталоне, от всех n-грамм сгенерированного реферата);

R – полнота (доля n-грамм в сгенерированном реферате, которые также были в эталоне, среди всех n-грамм эталона);

α = 0,9; β = 3,0; γ = 0,5; – коэффициенты;

frag – доля групп n-грамм (chunk) от количества n-грамм в сгенерированном реферате;

score – оценка METEOR.

Метрика **BERTScore** [42] использует для оценки качества контекстуальные эмбеддинги токенов предобученной модели BERT. Семантическая близость двух предложений вычисляется как косинусная близость между их эмбеддингами. Из всех метрик BERTScore показывает наилучшие результаты корреляции с человеческой оценкой.

Точность, полнота и F-мера по BERTScore [42]:

$$P_{BERT} = \frac{1}{|\hat{x}|}\sum_{\hat{x}_i \in \hat{x}} \max_{x_i \in x} \mathrm{x_i^\intercal}\,\hat{\mathrm{x}}_\mathrm{j},\ R_{BERT} = \frac{1}{|x|}\sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \mathrm{x_i^\intercal}\,\hat{\mathrm{x}}_\mathrm{j},\quad F_{BERT} = 2\frac{P_{BERT} \cdot R_{BERT}}{P_{BERT} + R_{BERT}},$$

где

$x$ – токенизированный эталонный реферат;

$\hat{x}$ – токенизированный сгенерированный реферат;

$\mathrm{x_i}$ – эмбеддинг из эталонного реферата;

$\hat{\mathrm{x}}_\mathrm{i}$ – эмбеддинг из сгенерированного реферата.

# Приложение С

## Примеры генерации рефератов

В квадратные скобки выделены отрывки текста, которые галлюцинируют (повторяются части текста).

Цветом выделены части текста, охватывающие аспекты статьи:

- Поставленная задача;
- Используемые методы;
- Полученные результаты.

| **Для длинного текста ВКР (5 586 токенов*)** | |
|---|---|
| Экспертный реферат | *ВКР посвящена задаче аспектного анализа научно-технических текстов. Предложены методы извлечения аспектных индикаторов с помощью лексико-синтаксических шаблонов. Описан процесс создания списка аспектных индикаторов и двух наборов шаблонов для языка LSPL. Созданный программный модуль был проверен на задаче автоматического реферирования текстов.* |
| ruT5-base-absum | Основы подхода к извлечению информации из текстов с помощью аспектных индикаторов были заложены в 70-е годы |
| rut5_base_sum_gazeta | Методы аспектного анализа научно-технических текстов на русском языке были разработаны в рамках выпускной квалификационной работы, посвященной разработке методов анализа текста. |
| rut5-small_g | Для анализа научно-технических текстов [было предложено сформировать список кандидатов в аспектные индикаторы] и опирающиеся на него методы аспектного анализа. |
| rut5-small_sa | В работе было предложено расширить аспектный анализ научно-технических текстов, которые характеризуют устойчивые слова, которые являются основными аспектными индикаторами для анализа текстов на русском языке. [В результате анализа научно-технических текстов было предложено рассматривать задачу автоматического реферирования текстов.] |
| rut5-small_gs | Для анализа научно-технических текстов в работе введен метод автоматического реферирования текстов. Для анализа научно-технических текстов было предложено рассматривать аспектные индикаторы[, которые характеризуют аспектные индикаторы], которые характеризуют аспекты анализа научно-технических текстов. |
| rut5-small_ga | Для анализа научно-технических текстов было предложено автоматически создать шаблоны для анализа научно-технических текстов. [Для этого было предложено сформировать список кандидатов в аспектные индикаторы], опирающиеся на него методы аспектного анализа и использовать уже существующую библиотеку для извлечения информации из текстов на естественном языке. Для этого было предложено [расширить полученные результаты] и уточнить полученные результаты. |
| rut5-small_gsa | В работе было создано программное средство, разработанное на основе выделения аспектных индикаторов для анализа научно-технических текстов на русском языке. [Для этого было предложено сформировать список кандидатов в аспектные индикаторы] и предложить опирающиеся на него методы автоматического реферирования текстов. [Для этого было предложено рассмотреть задачу автоматического реферирования текстов.] [Для снижения трудоемкости при его создании был предложен метод частичной автоматизации процесса.] |

*Полученных с помощью токенизатора razdel

| Для среднего текста статьи (2 779 токенов) | |
|---|---|
| Экспертный реферат | *Работа посвящена вариативности твердых/мягких согласных перед е в заимствованных словах. Целью данного проекта является создание обширной базы слов, в которых встречается эта вариативность.* |
| ruT5-base-absum | Проект создания базы слов с вариативным произношением согласных перед е. Основным назначением проекта является составление списка слов с вариативным произношением согласных перед е. Основным назначением проекта является сбор слова. |
| rut5_base_sum_gazeta | Впервые в истории русской речи появились слова с твердым и мягким произношением перед е. Это свидетельствует о том, что более частотные заимствованные слова сохраняют твердость. |
| rut5-small_g | Для того чтобы избавиться от нереализованных строк, можно почерпнуть из двух источников. Для того чтобы убедить исследователей, можно будет избавиться от ненужных строк, а также избавиться от ненужных строк. |
| rut5-small_sa | В работе С. В. Панов назвал цель создания базы заимствованных слов с вариативностью твердого/мягкого согласного, а в речи интеллигенции, чиновничества, высших классов, знающих [иностранные языки] |
| rut5-small_gs | Для того чтобы убедить в этом, что более частотность заимствованных слов в русском языке и его престижность в определенных группах информантов и иностранных языках можно выбрать список слов, которые можно было быть использованы для экспериментов. В результате эксперимента можно почерпнуть список слов, которые можно было быть использованы для экспериментов. |
| rut5-small_ga | Для создания базы заимствованных слов с вариативностью твердого/мягкого согласного перед е можно почерпнуть из двух источников. [Для этого было решено использовать два источника, которые можно почерпнуть из двух источников.] Для этого можно будет использовать новые версии орфоэпических словарей русского языка. Для этого можно будет использовать новые версии для исследований в области мягкого произнесения согласного. |
| rut5-small_gsa | В работе будет представлен проект создания базы заимствованных слов с вариативностью твердого/мягкого согласного перед е. В основу разметки было решено использовать два источника. В основу разметки являются множество факторов, релевантных для поведения согласного. [В основу разметки являются частотность и частотность для экспериментов], которые можно почерпнуть из двух источников, которые могут быть использованы для подготовки новых версий орфоэпических словарей русского языка. |
| **Для короткого текста тезисов (285 токенов)** | |
| Экспертный реферат | *Применение однослойной нейронной сети в задаче диагностики патологий кровеносной системы при поражении печени. В качестве обучающей выборки рассматривались результаты расчетов графов систем сосудов.* |
| ruT5-base-absum | Методы искусственного интеллекта к результатам математического моделирования кровеносной системы |
| rut5_base_sum_gazeta | Исследование возможности применения методов искусственного интеллекта к результатам математического моделирования кровеносной системы показало, что с помощью нейросети можно диагностировать заболевание по косвенным данным в сосудах, доступных для не инвазивного снятия параметров кровотока. |
| rut5-small_g | Система кровообращения, используемая для моделирования кровеносной системы, может рассчитывать параметры кровотока в любой точке системы сосудов человека, как в норме, так и патологии. Это может помочь совместить средства классического математического моделирования гемодинамики и средства искусственного интеллекта. |
| rut5-small_sa | В данной работе проведена исследование возможностей применения методов искусственного интеллекта к результатам моделирования явления печеночной гипертензии, [используемой для моделирования явления печеночной гипертензии], |

| | |
|---|---|
| | существенного изменения кровотока организма при поражениях печени, и средства искусственного интеллекта (нейронные сети) для взаимного дополнения |
| rut5-small_gs | Ученые рассказали, что используемая метод искусственного интеллекта для моделирования явления печеночной гипертензии, существенного изменения кровотока организма при поражениях печени, и патологии при различных степенях поражения печени, [используемая для моделирования кровеносной системы осуществлялась методом искусственного интеллекта,] |
| rut5-small_ga | Исследование возможностей применения методов искусственного интеллекта к результатам моделирования кровеносной системы осуществлено с помощью бинарной классификации. Это было проведено с помощью бинарной классификации для взаимного дополнения и развития. Это было проведено с помощью достаточно простой однослойной нейронной сети для взаимного дополнения и развития. Это было проведено с помощью программного комплекса CVSS. Это позволяет рассчитывать параметры кровотока в любой точке системы сосудов человека, как в норме, так и патологии. |
| rut5-small_gsa | [Исследование возможностей применения методов искусственного интеллекта к результатам моделирования кровеносной системы осуществлено в данной работе] с помощью достаточно простой однослойной нейросети. Создание этой сети, исследование ее свойств и реализации при различных функциях активации и значениях параметров для достаточно точного диагностирования. Для анализа можно диагностировать рассмотренную патологию кровотока по косвенным данным в сосудах, доступных для не инвазивного снятия параметров кровотока. |