Санкт–Петербургский государственный университет

Лямин Владимир Андреевич

ВКР бакалавра

Абстрактивная суммаризация русскоязычного текста

Уровень образования: бакалавриат

Направление 02.03.03 «Математическое обеспечение и администрирование информационных систем»

Основная образовательная программа СВ.5006.2018 «Математическое обеспечение и администрирование информационных систем»

Профиль «Администрирование информационных систем

Научный руководитель:
к.ф.-м.н., доц. Е.Г. Михайлова

Технический консультант:
В. А. Малых

Санкт-Петербург
2022 г.

Содержание

1. Введение

В настоящее время в мире распространена задача суммаризации текста. Суммаризация текста применяется при составлении анотации к научным статьям или к новостям, к краткому пересказу произведений и в других случаях.

Сейчас существуют два способа автоматической суммаризации текста. Один из них экстрактивный[1], второй абстрактивный[2]. Расскажем о каждом из них поподробнее.

Экстрактивная суммаризация является наиболее простым видом для реализации и понимания. Ее принцип заключается в выделении из текста наиболее важных информационных блоков. Каждый блок может являться абзацем, предложением и так далее[3]. Работа данного подхода показана на рисунке 1.

Абстрактивная суммаризация является более сложным подходом для выделения главной мысли из текста. Она заключается в создании собственного краткого содержания, которое может содержать слова, которые не имеет исходный текст. Работа данного подхода показана на рисунке 2.

Если подвести итог, то оба подхода имеют как минусы, так и плюсы:

- экстрактивный подход более прост для изучения;
- экстрактивный подход более прост для реализации;
- абстрактивный подход выполняет свою работу более качественно.

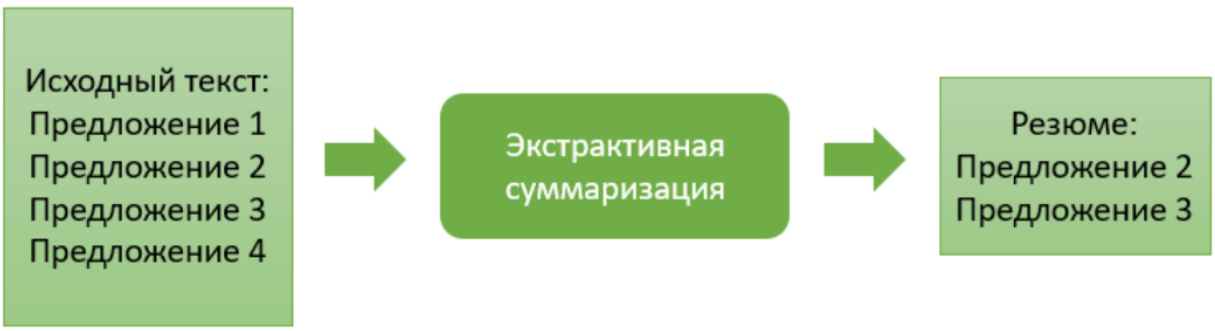

Рис. 1: Абстрактивная суммаризация

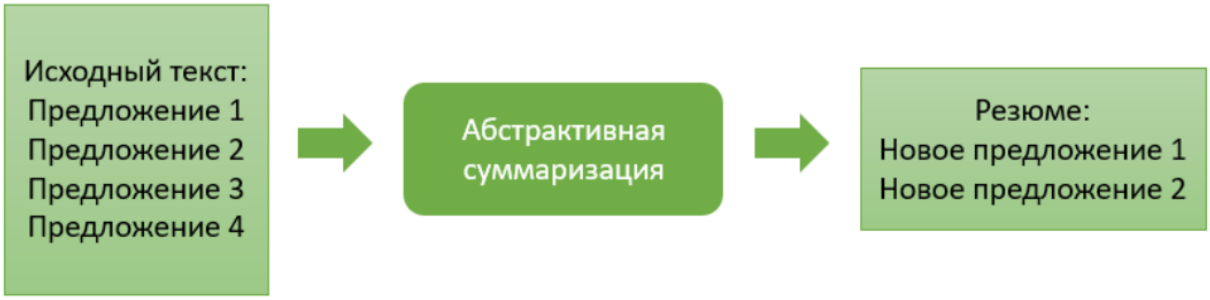

Рис. 2: Экстрактивная суммаризация

На данный момент наиболее развиты системы по суммаризации англоязычных текстов. Существует множество обученных моделей на множествах данных, таких как CNN, Reddit, XSum и так далее. Тогда как решений для русскоязычных текстов не так много, как и самих множеств данных.

В этой связи возникла потребность в успешном обучении модели для суммаризации русскоязычного текста с приемлимым Rouge. Далее будем говорить о задачах, которые были поставлены непосредственно в данном проекте.

2. Постановка задачи

Целью данной работы является создание качественного решения для абстрактивной суммаризации русскоязычного текста. Для достижения этой цели были сформулированы следующие задачи.

2.1 Задачи

- Изучить существующие модели экстрактивной суммаризации.
- Воспроизвести результаты исследователей и провести тестирование с другим набором данных.
- Изучить решения для абстрактивной суммаризации русскоязычного текста и воспроизвести результаты.
- Самостоятельное обучение модели абстрактивной суммаризации на русскоязычном наборе данных.
- Тестирование на других наборах данных.

3. Существующие модели экстрактивной суммаризации

В ходе исследования были рассмотрены несколько моделей экстрактивной суммаризации. Такие как MatchSum [4], DiscoBERT, BertSumExt. Рассмотрим каждую из них более подробно.

MatchSum - это система основанная на Siamese BERT архитектуре. Для сопоставления документа D и резюме кандидата C, исследователи использовали исходный BERT, чтобы получить семантически значимые вложения из документа D и резюме кандидата C. Основная идея модели состоит в том, чтобы дать золотому резюме наивысшую оценку соответствия, и в то же время лучшее резюме кандидата должно получить более высокий балл по сравнению с неквалифицированным резюме кандидата.

DiscoBert извлекает токены в качестве кандидатов для экстрактивного отбора с более высокой степенью детализации. Чтобы устоялись долгосрочные зависимости, структурные графы строятся на основе деревьев RST. Авторы показывают, что предлагаемая модель превосходит современные методы по популярным эталонным тестам суммирования по сравнению с другими моделями на основе BERT.

BertSumExt представляет собой предварительно обученные языковые модели. Авторы представляют новый кодировщик, основанный на BERT, который способен выражать семантику документа и получать представления для его предложений. Данная модель извлечения построена поверх кодировщика путем наложения нескольких слоев Transformer между предложениями.

Для сравнения систем было решено использовать метрику ROUGE, так как она является наиболее объективной оценкой качества суммаризации. В ходе анализа данных моделей был выбран MatchSum, так как он превосходит свои аналоги по характеристикам (таб. 1). Также MatchSum имеет более понятное руководство, используя которое можно воспроизвести результаты и протестировать систему на собственных данных.

Рассматривались и другие аналоги этих систем. Но они имеют более низкую эффективность по сравнению с этими тремя моделями, а также

они является более старыми системами.

X	MatchSum	DiscoBert	BertSumExt
R-1	44.41	43.77	43.85
R-2	20.86	20.85	40.67
R-l	40.55	20.34	39.90

Таблица 1: Заявленные результаты авторов моделей.

4. Тестирование MatchSum

4.1 Воспроизведение результатов

Для вопроизведения результатов работы MatchSum требуется операционная система Linux, поэтому было принято решение использовать Google Colab. Так как данная система позволяет писать код на языке Python, а также позволяет бесплатно пользоваться своими вычислительными ресурсами.

Перед началом тестирования необходимо установить дополнительные библиотеки, такие как: fastNLP [5], PyTorch [6]... После этого нужно скачать базы данных. Авторы дают уже подготовленные данные для тестирования на выбор. Это либо CNN, либо DailyMail.

Так как время жизни виртуальной машины в Google Colab не превышает двенадцати часов, а тестирование составляет порядка тридцати часов, то потребуется подредактировать код программы, указанным авторами способом[7]. Далее можно запускать процесс обучения модели. Тут авторы дают возможность выбрать модель для обработки естественного языка. Это BERT или RoBERTa. После этого можно запускать процесс тестирования. В итоге получился результат близкий к исходному результату (рис. 3).

Model	R-1	R-2	R-L
MatchSum (BERT-base)	44.22	20.62	40.38
MatchSum (RoBERTa-base)	44.41	20.86	40.55

Рис. 3: Результат MatchSum

4.2 Тестирование MatchSum с русскоязычным набором данных

Русскоязычных наборов данных, находящихся в открытом доступе, доступно небольшое количество. Это Gazeta, Lenta, RIA. Для данной модели была выбрана база данных RIA[?]. Для того, чтобы программа заработала необходимо трансформировать json файл в нужный формат. Документ должен содержать поля "text"и "summary". В качестве "text"был выбран сам текст новости, а в качестве "summary"был выбран ее заголовок. Чтобы удовлетворить этим условиям были написаны необходимые программы. Затем нужно пропустить текст через систему экстрактивной суммаризации Bert. Эти данные подаются на вход в программу MatchSum. Результат работы программы показан на рисунке 4.


```
{"text": ["около 60 тысяч литров неочищенной нефти вытекли из нефтепровода
компании    на территории северной канадской провинции альберта в
результате механического повреждения, сообщил в воскресенье регулятор в
области энергетики альберты.", "согласно информации на сайте регулятора,
механическое повреждение вызвало утечку нефти, равноценную нескольким
сотням баррелей, в болото.", "инцидент произошел в минувший четверг в 350
километрах к северу от административного центра провинции  эдмонтона.",
"по данным представителя регулятора кэрри росы, нефтяная компания подала
заявление после инцидента.", "из-за вопросов безопасности и дорожных
условий мы не смогли направить инспекторов к месту происшествия.", "как
только это станет безопасно, мы сделаем это,  цитирует росу агентство
..", "рейтер.", "сообщается, что на местности началась очистка
загрязнения."
], "summary": ["в канаде из нефтепровода в болото вытекли около 60 тысяч
литров нефти"
```


Рис. 4: Результат работы программы

5. Обученные модели для абстрактивной суммаризации русскоязычного текста

В ходе проведения исследования также были рассмотрены и произведены результаты с уже натренированными моделями абстрактивной суммаризации. Это mBART и Pointer-Generator.

Модель BART - это sequence-to-sequence модель, которая обучалась на воссоздании испорченного текста. В отличие от своего аналога BERT данная система сразу обучалась на генерации текста, поэтому для автоматического реферирования подходит лучше. mBART – это многоязычная версия BART[8].

Модель Pointer-Generator является гибридной сетью, которая может копировать слова из источника с помощью указания, сохраняя при этом возможность генерировать слова из постоянного словаря[9].

Автор дает на выбор два набора данных. Можно использовать набор данных либо Gazeta, либо RIA. Результаты представлены в таблице 2.

X	mBART	Pointer-Generator
R-1	32.4	26.4
R-2	14.3	9.8
R-l	28	22.4

Таблица 2: Результаты работы моделей на множестве данных Gazeta.

6. Обучение модели абстрактивной суммаризации на русскоязычном наборе данных

6.1 Архитектура

Архитектура модели основывается на se2seq модели Xin Pan and Peter Liu[10]. Кодировщик состоит из двухслойной двунаправленной RNN с LSTM. Была выбрана двунаправленная система, так как так модель может изучать более сложные шаблоны. Декодеровщик состоит из двухслойной однонаправленной RNN с LSTM с использованием Bahdanau attention[11]. Bahdanau attention был использован для более быстрого и качественного обучения модели. Архитектуру можно увидеть на рисунке 5.

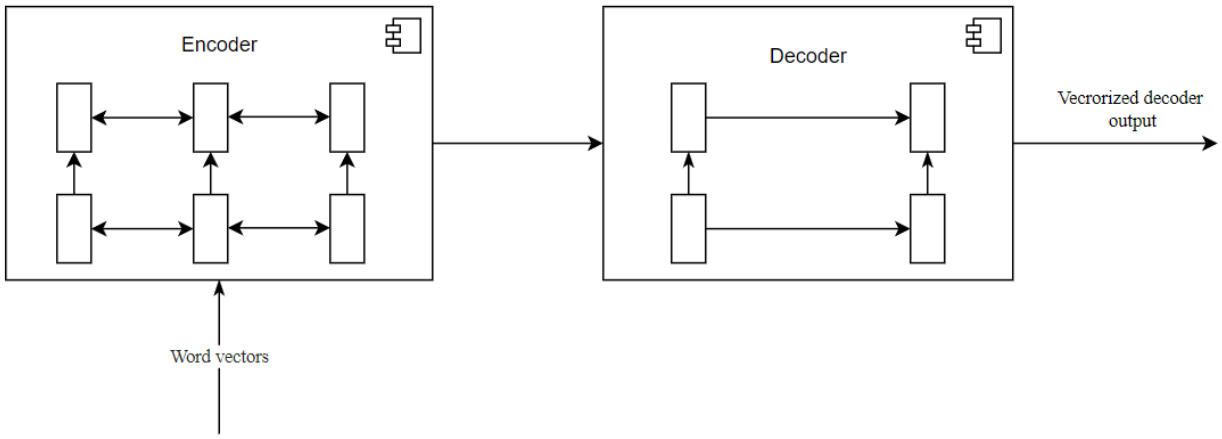

Рис. 5: Диаграмма компонентов модели

6.2 Используемые технологии и данные

Перед началом обучения модели необходимо продумать на каких мощностях данный процесс будет происходить. В качестве систем рассматривались два аналога. Первой системой был Google Colab второй Yandex DataSphere[12]. У каждой системы есть свои достоинства и недостатки. Рассмотрим их более подробно.

К положительной стороне Google Colab можно отнести простоту его использования и бесплатное использование их вычислительных мощностей.

К отрицательной относятся ограничение на использование памяти, небольшое время жизни виртуальной машины, после истечения которой все данные стираются.

В Yandex DataSphere возможно хранить данные достаточно большого объема необходимые для процесса обучения и тестирования, так как жизнь виртуальной машины неограничена. Также есть возможность регулировать необходимые вычислительные ресурсы. К недостаткам относится то, что данная система требует определенную плату за предоставление своих мощностей, а также более сложна в использовании.

В качестве библиотеки для автоматической суммаризации была выбрана TensorFlow. Она была выбрана, так как это одна из лучших библиотек для машинного обучения. Также для нее написано большое количество подробной документации и она установлена по умолчанию в Yandex DataSphere.

Далее необходимо выбрать набор данных, на которых будет происходить обучение. Из трех имеющихся видов данных были выбраны данные от gazeta. Это было сделано, так как на них получался наиболее высокий результат.

6.3 Процесс обучения модели

Для начала необходимо подготовить данные. Данные из формата json были преобразованы в формат csv. Далее нужно выбрать какие поля будут являтся исходным текстом, а какие справочной сводкой. Было решено, чтобы текстом служил сам тест статьи, а справочной сводкой был заголовок этой статьи. Все остальные поля удаляются.

Следующим этапом будет удаление пунктуации из текста, а также удаление стоп-слов из текста, так как они только мешают правильному обучению модели[13]. Также это ускорит процесс обучения, так как уменьшит количество данных.

В качестве словаря используемых слов были использованы данные из ConceptNet Numberbatch для русского языка, а также слова, которые встречаются более десяти раз в наборе используемых данных от gazeta. Из

этих слов строится эмбеддинг слов.

Для построения уровня кодирования, был использован двунаправленный RNN с LSTM, а для уровня декодирования двухслойный LSTM. Матрица эмбеддингов слов, построенная ранее, используется на обоих данных процессах.

Далее строится граф и по нему происходит обучение модели. Модель была обучена на множестве из 30000 статей. Результаты тестирования можно увидеть в таблице 3.

R-1	R-2	R-l
33.6	15.5	29.1

Таблица 3: Результаты работы модели на множестве данных Gazeta.

7. Тестирование на других наборах данных

В качестве эксперимента было решено взять данные от Lenta. В результате обучения модели на этих данных и последующего тестирования получились следующие результаты, которые можно увидеть в таблице 4.

R-1	R-2	R-l
31.5	14.8	26.4

Таблица 4: Результаты работы модели на множестве данных Lenta.

8. Заключение

В ходе данной работы были достигнуты следующие результаты.

- Изучены существующие модели экстрактивной суммаризации на примере MatchSum.
- Воспроизведены результаты исследовний MatchSum и проведены тесты с русскоязычным текстом.
- Изучены решения для абстрактивной суммаризации русскоязычного текста и воспроизведены результаты.
- Была обучена собственная модель абстрактивной суммаризации при помощи библиотеки TensorFlow на данных от Gazeta.
- Проведено тестирование с набором данных от Lenta.

Код для обучения модели представлена на GitHub: https://github.com/VLiamin/AbstractiveModel.

Список литературы

[1] Understanding automatic text summarization-1: Extractive methods. *https://towardsdatascience.com/understanding-automatic-text-summarization-1-extractive-methods-8eb512b21ecc*, 2022-May(2).

[2] Abstractive summarization: An overview of the state of the art. *https://www.sciencedirect.com/science/article/abs/pii/S0957417418307735*, 2022-May(2).

[3] Суммаризация текста: подходы, алгоритмы, рекомендации и перспективы. *https://habr.com/ru/post/514540/*, 2021-November(28).

[4] Extractive summarization as text matching. *https://arxiv.org/pdf/2004.08795.pdf*, 2021-November(28).

[5] fastnlp. *https://github.com/fastnlp/fastNLP*, 2021-November(28).

[6] From research to production. *https://pytorch.org/*, 2021-November(28).

[7] Matchsum. *https://github.com/maszhongming/MatchSum*, 2021-November(29).

[8] Секреты генерирующего реферирования текстов. *https://habr.com/ru/post/596481/*, 2022-May(2).

[9] Taming recurrent neural networks for better summarization. *http://www.abigailsee.com/2017/04/16/taming-rnns-for-better-summarization.html*, 2022-May(2).

[10] Abstractive text classification using sequence-to-convolution neural networks. *https://arxiv.org/abs/1805.07745*, 2022-May(2).

[11] The bahdanau attention mechanism. *https://machinelearningmastery.com/the-bahdanau-attention-mechanism/*, 2022-May(2).

[12] Как начать работать с datasphere. *https://cloud.yandex.ru/docs/datasphere/qui* 2022-May(2).

[13] Text pre-processing: Stop words removal using different libraries. *https://towardsdatascience.com/text-pre-processing-stop-words-removal-using-different-libraries-f20bac19929a*, 2022-May(2).