САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Математико-Механический факультет
Кафедра информационно аналитических систем

Суммаризация групп в социальных сетях

Дипломная работа студента 645 группы Чурикова Никиты Сергеевича

Научный руководитель:
к.ф. - м.н., доцент Графеева Н. Г.

Рецензент:
Руководитель департамента вычислительной биологии
Яковлев П. А.

Заведующий кафедрой:
к.ф. - м.н., доцент Михайлова Е. Г.

Санкт-Петербург
2019 г.

Содержание

1 Аннотация

Одной из задач обработки естественного языка является задача суммаризации текста. Ее целью является уменьшение размера исходного текста без потери ключевой информации. В данной работе мы решаем схожую проблему, но для информационных ресурсов в социальных сетях. В частности, необходимо рассмотреть задачу суммаризации текстов и картинок, поскольку это два основных источника информации. В тексте мы приводим численное обоснование выбранных методов, а также приводим оценку нашей суммаризации людьми.

2 Введение

В современном мире создается все больше и больше информации, которую мы можем потреблять. Новости, статьи, юмор постоянно меняются и создаются людьми. При таком потоке информации появляется потребность в инструментах, способных давать как можно больше информации с минимальными потерями.

При чтении новостей люди, как правило, не идут дальше новостных заголовков [5], для популярных технических статей создают краткие описания описывающие их достижения и основные моменты [19, 2], а визуальный контент нередко подчиняется единому шаблону.

В данной работе мы показываем, как используя современные достижения в области анализа данных можно извлекать полезную информацию из новостных ресурсов в социальной сети вконтакте [21], и приводим обоснование выбора решений, основываясь на соответствующих метриках.

2.1 Постановка задачи

Мы поставили перед собой задачу создать систему, которой бы можно было передавать ссылку на новостной ресурс в социальной сети вконтакте, а на выходе получать его краткое описание. В рамках работы мы ограничились новостными ресурсами с высоким содержанием текста.

На Рис. 1 мы показываем как работает наше решение "с высоты птичьего полета". Процесс имеет следующий вид, мы получаем ссылку на группу ВК, затем через API вконтакте получаем информацию о группе и оцениваем количество текста в ней. если содержание текста низкое, то мы говорим, что это ресурс с доминирующим медиа-контентом, если в группе мало предложений, но достаточно слов, то решается задача выделения ключевых слов, если в группе высокое содержание предложений, то мы решаем задачу генерации заголовков.

Таким образом, с алгоритмической точки зрения, задача суммаризации новостного ресурса была рассмотрена нами как две подзадачи:

1. Извлечение ключевых слов, присущих данному источнику информации;

2. Сжатие новостей, используя автоматическое создание заголовков.

Через извлечение данной информации мы хотим добиться эффекта "чтения по диагонали".

Для оценки качества наших алгоритмов, мы воспользовались открытыми датасетами для генерации заголовков и извлечению ключевых слов.

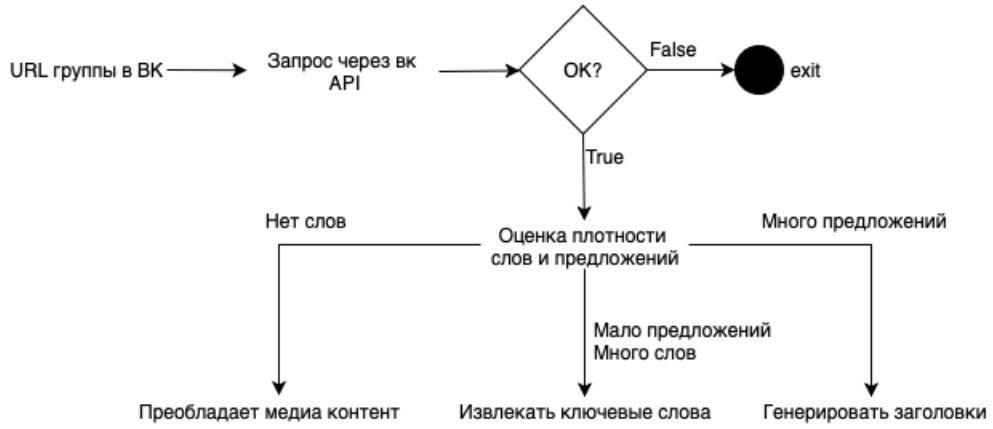

Рис. 1: Принцип работы системы.

2.2 Обзор литературы

Задача сжатия текста с малой потерей смысла и сохранением возможности его прочтения имеет название задачи *суммаризации*. При этом, есть два концептуальных подхода к решению: экстрактивный, когда для создания краткого содержания извлекаются целые куски текста вплоть до предложений, и абстрактивная, где в кратком содержании могут быть слова, которых не было в исходном тексте.

В частности, при исследовании абстрактивной генерации заголовков, мы отталкивались от статьи Вконтакте, посвященной данной проблеме [6]. Ими предлагается применять нейронные сети с архитектурой Transformer и предобработкой Byte pair encoding (BPE) [18]. Однако в задаче абстрактивной генерации заголовков существуют дебаты на тему того, что использовать в качестве входа модели. Поскольку долгое время SOTA были модели с архитектурой encoder decoder, то было невозможно использовать длинные входные последовательности. Потому авторы статьи [14] исследуют различные подходы по предварительному извлечению "Topic sentence которое нейронная сеть должна дальше обработать. Это предложение, как говорят авторы, в идеальном случае, должна отвечать на 5W1H. Но достаточно ответов на "что, кто, когда".

Для экстрактивной суммаризации чаще всего используют алгоритм TextRank [12].

3 Алгоритмы, использованные в работе

Нами были использованы как классические подходы, так и новые, основанные на нейронных сетях. В следующих секциях мы опишем их основные принципы, а также приведем ссылки на их реализации.

3.1 Суммаризация текста

Для суммаризации текста мы воспользовались алгоритмом экстрактивной суммаризации основанном на TextRank [20, 15, 12], и моделью трансформера [3], обученной на датасете РИА новостей [6]. Для предобработки данных модели трансформера мы использовали byte pair encoding [18]. Помимо этого мы извлекали первое предложение из новости. Для TextRank и извлечения первого предложения не требуется обучающая выборка, что делает их очень удобными в использовании. При этом, исследования показывают, что в задаче генерации заголовков, первое предложение в новости – это очень сильный бэйзлаин [6], который трудно побить как экстрактивной, так и абстрактивной суммаризацией.

3.1.1 Baseline

В качестве бэйзлаина в задаче генерации заголовков используется первое предложение новости. Именно им мы и воспользовались и отталкивались от него.

3.1.2 TextRank

TextRank является является адаптацией идеи алгоритма PageRank [13] с задачи рекомендации страниц в интернете на задачу рекомендации лучшего предложения или набора слов в тексте. Сам алгоритм состоит в том, что мы текст превращаем в граф, где узлы – это предложения, а для каждого ребра подсчитывается вес, где вес определяется по количеству совпавших слов в двух предложениях.

Таким образом, получается, что можно выбрать предложение с самыми тяжелыми ребрами в качестве предложения, которое описывало бы исходный текст.

3.1.3 Byte pair encoding

Мы используем byte pair encoding (BPE), технику, предложенную Сеннрич для задачи машинного перевода в [18]. BPE – это метод сжатия данных, в котором часто встречающиеся пары байтов заменяются дополнительными символами алфавита. В случае текстов, как в области машинного перевода, наиболее часто встречающиеся слова сохраняются в словаре, а менее часто встречающиеся слова заменяются последовательностью (обычно двумя) токенами. Например, для морфологически богатых языков окончания слов могут быть отделены, поскольку каждая форма слова определенно реже, чем ее основа. Кодирование BPE позволяет нам представлять все слова, включая те, что не встречаются по время обучения, с фиксированным словарным запасом.

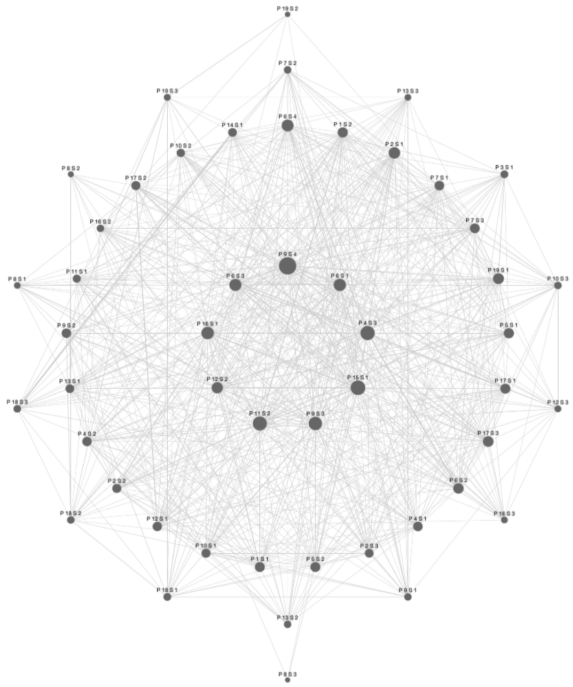

Рис. 2: Пример результирующего графа textrank.

3.1.4 Universal transformer network

В то время как рекуррентные нейронные сети могут быть легко использованы для определения модели Encoder-Decoder, тренировка таких моделей очень дорого с точки зрения вычислений. Другой недостаток состоит в том, что они используют только локальную информацию, опуская последовательность скрытых состояний H = h1, ..., hN. То есть любые два вектора из скрытого состояния hi и hj связаны с вычислениями j - i RNN, что затрудняет улавливание всех зависимостей в них из-за ограниченной емкости. Чтобы обучить богатую модель, которая изучила бы сложную текстовую структуру, мы должны определить модель, которая опирается на нелокальные зависимости в данных. В этой работе мы принимаем архитектуру модели Universal Transformer [], которая является модифицированной версией Transformer []. Этот подход имеет несколько преимуществ по сравнению с RNN. Прежде всего, его можно тренировать параллельно. Кроме того, все входные векторы связаны друг с другом через механизм Attention. Это подразумевает, что архитектура Transformer учитывает нелокальные зависимости между токенами независимо от расстояния между ними, и, таким образом, она может выучить более сложное представление текста в статье, что оказывается необходимым для эффективного решения задачи суммаризации.

3.2 Выделение ключевых слов

Обзорная статья Сайдула и Ына [8] рассматривает наиболее сильные подходы к выделению ключевых слов. Конкретно, они показывают для каких задач доходят какие алгоритмы. Основная таблица этой статьи приведена на таблице 1, и она говорит о том, какие алгоритмы показывают лучшие результаты на каких задачах.

Для данной работы мы реализовали TopicRank и TextRank и выложили эти реализации в качестве библиотеки на python [1], реализовав интерфейсы библиотеки Scikit-Learn [17, 1].

Dataset	**Approach and System**	**P**	**R**	**F**
Абстракты	TopicRank	35.0	66.0	45.7
Блоги	CommunityCluster	35.1	61.5	44.7
Новости	TextRank	28.8	35.4	31.7
Научные статьи	Statistical, semantic, and distributional features	27.2	27.8	27.5

Таблица 1: Sate of the art результаты извлечения ключевых слов на классических датасетах. P – точность, R – полнота, F – среднее гармоническое точности и полноты.

В секциях ниже мы опишем детали наших реализаций данных алгоритмов.

3.2.1 TopicRank

TopicRank - это метод обучения без учителя, целью которого является извлечение ключевых фраз из наиболее важных тем документа. Темы определяются как кластеры похожих ключевых фраз-кандидатов. Извлечение ключевых фраз из документа состоит из следующих шагов, показанных на рисунке 3. Во-первых, документ предварительно обрабатывается (сегментация предложений, разметка слов и тегирование частей речи), а кандидаты в ключевые фразы группируются по темам. Затем темы ранжируются в соответствии с их важностью в документе, и ключевые фразы извлекаются путем выбора одного кандидата ключевой фразы для каждой из наиболее важных тем.

3.3 Оценки качества

Для оценки качества текстовых моделей мы использовали метрику ROUGE-L F1 [10], при этом мы считали ее на датасете РИА новостей [6]. Помимо этого, на основе этого датасета проводилось соревнование по генерации

[1] https://github.com/kuparez/keyverbum

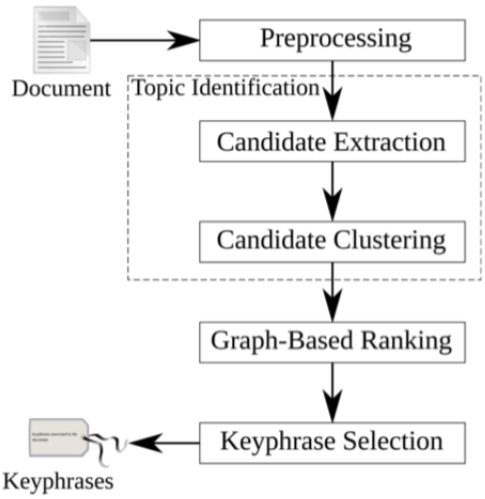

Рис. 3: Принцип работы TopicRank.

заголовков, где автором было получено 3 место, а описание результатов соревнования было подано в качестве статьи на конференцию "Диалог".

4 Анализ использованных данных

В работе были использованы несколько датасетов с целью выбора алгоритмов под соответствующие задачи. В частности, мы воспользовались датасетом оценки качества выделения ключевых слов [11], недавно опубликованным датасетом Риа новостей [6], а также в нашей работе мы используем данные из 25 групп вконтакте с разным целевым материалом: от медиа контента до полноценных длинных текстов.

4.1 Данные выделения ключевых слов

Для анализа качества использованных алгоритмов для выделения ключевых слов мы воспользовались датасетом с разнообразными статьями на русском языке [11]. В наборе присутствуют научные статьи из "журнала киберленика"[4], технического блога "хабрахабр"[7] и новостных ресурсов "Независимая газета"[9] и "Россия сегодня"[16].

Таким образом, мы приводим русскоязычные аналоги всем англоязычным ресурсам из таблицы 1.

5

For all experiments, we used the preprocessing as described above. For training of our models we used 8 Nvidia K80.

5.1 Evaluation

It is common practice to use ROUGE score [10] in summarization problems. In practice, so-called ROUGE 1,2, L - precision, recall, F1 metrics are being used. The names in ROUGE X - Y denote the following: X is the number of ngramm used to calculate the Y metric. In the case of ngramm of size L, the longest common subsequence from predicted sequence found in original sequence. Y metrics are classic accuracy, recall and F1 score.

In the competition, for the calculation of quality, the average of ROUGE 1, 2, L - F1 score was used.

5.2 First sentence

As described in [14], the first sentence for automatically creating a news headline is a very strong baseline. In particular, in the headline generation competition, the first sentence of news articles was proposed as a baseline solution.

However, we used our knowledge about the data and besides removal of html tags and entities we've also skipped first sentences with "риа новости"in it. This allowed us to skip sentences with information about the date the news was posted as these sentences did not contain any information besides that.

5.3 Textrank

We used the classic extractive summarization algorithm - textrank [20]. We took implementation of this algorithm from gensim [15] To create a summary using keywords and original sentences we've set extraction size to be 20% of the original text.

5.4 Transformer

Since the competition was held for only a month, we've decided to repeat the results described in the article VKontakte [6]. We took the implementation of the transformer from Open NMT, with the parameters described in the article.

Then, we tried to focus on what to use as input for the model. We tried the first sentence according to the rules above. We've also tried as an input first 2000 BPE tokens.

Due to restrictions on VK servers we could not make very complex predictions. There for, during predictions on leaderboard we've used beam search of size 5. But for evaluation on our test set we used beam size of 10.

6 Results

As could be seen in Table 1, none of the models we've trained could show better results than the preprocessed first sentence. Unfortunately, we could not check some of our models on VK servers due to the restrictions on time complexity of our models. For this reason, we also present results on our test set.

Algorithm	Score
Baseline	0.19500071
First sentence	0.19502427
Wiki BPE transformer (first sentence, no beam search=5)	0.16397515
Ria BPE transformer (2000 first tokens, beam search=5)	0.16131584
Textrank summarization	0.10764881
Textrank keywords	0.06259589

Таблица 2: Results from evaluation by VK servers. Score is mean of ROUGE-1,2,L F1 score.

Table 2 show results of evaluation on our test set. Surprisingly, we've got much better results on our split, than on public leaderboard. And specifically, neural networks perform much better than first sentence model. It's worth pointing out that transformer trained with RIA BPE perform slightly worse than transformer with Wiki BPE. We think, the reason for that might be that model is more robust, because it doesn't need to predict any variation of numbers, because all numeric data is reduced to zeros. Unfortunately, taxtrank based model didn't show good results for F1 score, but recall is pretty high. It's probably due to extraction of too much information.

Another thing is that local evaluation show better results than VK server evaluation. So we decided to check for data leaks. We assumed that there could be none, because for traintest split we used scikit learn [17] train test split method. We checked, how many texts in our test set are identical to texts in train. It turned out that 975 texts in test set were found in train set.

Then we decided to check original data and find how many training examples are identical. We found, that 2651 texts are identical.

After our discovery of a leak in our test set distribution, we decided to check performance of our best abstractive summarization model, transformer with Wikipedia BPE with beam search equal to 5 on dataset with train set leak removed.

As expected, model performed slightly worse, but still pretty good, so it's still hard to tell, why there is such a great difference between results on out test set and evaluation results from competition servers.

In table 4,the mean of ROUGE-1,2, L F1 is used as the score. In the table, we present two of the worst headline options that Wiki transformer can produce on data without a train leak in test.

Algorithm\Score	1F	1P	1R	2F	2P	2R	LF	LP	LR
First sentence	0.23	0.16	0.44	0.10	0.07	0.21	0.16	0.15	0.40
Wiki BPE transformer (first sentence, beam=10)	0.37	0.39	0.36	0.20	0.21	0.19	0.34	0.37	0.34
Wiki BPE transformer (first sentence, beam=5)	0.39	0.41	0.39	0.22	0.23	0.22	0.37	0.39	0.37
Ria BPE transformer (2000 first tokens)	0.36	0.37	0.35	0.18	0.20	0.18	0.33	0.36	0.33
Textrank summarization	0.14	0.09	0.41	0.05	0.03	0.17	0.09	0.08	0.38
Textrank keywords	0.09	0.07	0.18	0.00	0.00	0.01	0.05	0.05	0.14

Таблица 3: ROUGE-1,2,F1, precision and recall scores.

Algorithm\Score	1F	1P	1R	2F	2P	2R	LF	LP	LR
Wiki BPE transformer (first sentence, beam=5 with leak)	0.39	0.41	0.39	0.22	0.23	0.22	0.37	0.39	0.37
Wiki BPE transformer (first sentence, beam=5 no leak)	0.39	0.40	0.39	0.22	0.23	0.22	0.36	0.38	0.37

Таблица 4: Model evaluation without leak

Score	Example
0.00	**Text**: 8 декабря 1991 года россия, белоруссия и украина подписали соглашение о создании содружества независимых государств (снг). **Ground truth**: встреча в беловежской пуще **Prediction**: заявил
0.00	**Text**: более 70 международных художников и арт-групп примут участие в основном проекте "больше света" 5-й московской биеннале современного искусства,который будет показан в цвз "манеж" с 20 сентября по 20 октября, сообщили риа новости в пресс-службе проекта. **Ground truth**: стали известны все участники основного проекта 5-й московской биеннале **Prediction**: более 00 художников представят проект "больше света"в "манеже"
0.99	**Text**: фонд "сколково"и корпорация intel подписали в четверг соглашение о сотрудничестве, передает корреспондент риа новости. **Ground truth**: "сколково"и intel подписали соглашение о сотрудничестве **Prediction**: "сколково"и intel подписали соглашение о сотрудничестве

Таблица 5: Worse and best model performances. Score is mean of ROUGE-1,2,L F1 score.

7 Заключение

В данной работе мы предложили решение задачи краткого описания новостного ресурса. Решение включает в себя путь того, как с этой системой будет взаимодействовать пользователь, техническую архитектуру системы, обоснование выбранных алгоритмов, отталкиваясь от соответствующих решаемых подзадач.

Как результат, с алгоритмической точки зрения, система была разбита на две подзадачи: выделения ключевых слов и генерации заголовков. Нами были предложены лучшие алгоритмы, для решения соответствующих задач. В работе не был покрыт графический материал ввиду того, это является отдельным большим исследованием, при этом в этой работе мы заложили фундамент для дальнейших исследований в этой области, предоставив интерфейс для разработчиков и людей.

8 Литература

Список литературы

[1] API design for machine learning software: experiences from the scikit-learn project / Lars Buitinck, Gilles Louppe, Mathieu Blondel et al. // ECML PKDD Workshop: Languages for Data Mining and Machine Learning. — 2013. — P. 108–122.

[2] article essence. Article essence. — 2019. — feb. — https://opendatascience.slack.com/messages/C5VQ222UX.

[3] Attention is all you need / Ashish Vaswani, Noam Shazeer, Niki Parmar et al. // CoRR. — 2017. — Vol. abs/1706.03762.

[4] Cyberlenica. — https://cyberleninka.ru/.

[5] DeMers, J. 59 percent of you will share this article without even reading it. — 2016. — aug. — https://www.forbes.com/sites/jaysondemers/2016/08/08/59-percent-of-you-will-share-this-article-without-even-reading-it/#71991fa2a648.

[6] Gavrilov, D. Self-attentive model for headline generation / Daniil Gavrilov, Pavel Kalaidin, Valentin Malykh // Proceedings of the 41st European Conference on Information Retrieval. — 2019.

[7] Habr. — https://habr.com/ru/.

[8] Hasan, K. S. Automatic keyphrase extraction: A survey of the state of the art / Kazi Saidul Hasan, Vincent Ng // Proceedings of the

52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). — Baltimore, Maryland: Association for Computational Linguistics, 2014. — June. — P. 1262–1273. — http://www.aclweb.org/anthology/P14-1119.

[9] Independent journal. — http://www.ng.ru/.

[10] Lin, C.-Y. Rouge: A package for automatic evaluation of summaries / Chin-Yew Lin // Proc. ACL workshop on Text Summarization Branches Out. — 2004. — P. 10. — http://research.microsoft.com/~cyl/download/papers/WAS2004.pdf.

[11] Mannefedov. ru-kw-eval-datasets. — 2019. — Jan. — https://github.com/mannefedov/ru_kw_eval_datasets.

[12] Mihalcea, R. Graph-based ranking algorithms for sentence extraction, applied to text summarization / Rada Mihalcea. — 2004. — 01. — Vol. 170-173.

[13] Page, L. The pagerank citation ranking: Bringing order to the web. — 1998.

[14] Putra, J. W. G. Incorporating topic sentence on neural news headline generation / Jan Wira Gotama Putra, Hayato Kobayashi, Nobuyuki Shimizu. — 2018.

[15] Řehůřek, R. Software Framework for Topic Modelling with Large Corpora / Radim Řehůřek, Petr Sojka // Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. — Valletta, Malta: ELRA, 2010. — May. — P. 45–50. — http://is.muni.cz/publication/884893/en.

[16] Rt in russian, latest news in the world and russia. — https://russian.rt.com/.

[17] Scikit-learn: Machine learning in Python / F. Pedregosa, G. Varoquaux, A. Gramfort et al. // Journal of Machine Learning Research. — 2011. — Vol. 12. — P. 2825–2830.

[18] Sennrich, R. Neural machine translation of rare words with subword units / Rico Sennrich, Barry Haddow, Alexandra Birch // CoRR. — 2015. — Vol. abs/1508.07909.

[19] tldr arxiv. tldr arxiv. — 2019. — feb. — https://t.me/tldr_arxiv.

[20] Variations of the similarity function of textrank for automated summarization / Federico Barrios, Federico López, Luis Argerich, Rosa Wachenchauzer // CoRR. — 2016. — Vol. abs/1602.03606.

[21] vkontakte. vkontakte. — 2019. — feb. — https://vk.com/feed.