САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Математико-Механический факультет

Кафедра информационно аналитических систем

Суммаризация групп в социальных сетях

Дипломная работа студента 646 группы
Чурикова Никиты Сергеевича

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

Рецензент:
Директор департамента управления информационными потоками Яковлев П. А.

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

Санкт-Петербург
2019 г.

SAINT PETERSBURG STATE UNIVERSITY

Mathematics and mechanics department
Sub-Department of Analytical Information System

News resources summarization in social networks

Final thesis of student of 646 group
Churikov Nikita Sergeevich

Scientific Supervisor:
Assistant Professor Grafeeva N. G.

Reviewer:
Director of information flow department Yakovlev P. A.

Sub department director:
Assistant Professor Michailova E. G.

Saint Petersburg
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, мы рассматриваем задачи генерации заголовков и выделения ключевых слов, поскольку тексты бывают разного объема и потому их можно сжимать разными способами. В тексте мы приводим численное обоснование выбранных методов и описываем интерфейс разработка.

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 [6], для популярных технических статей создают краткие описания описывающие их достижения и основные моменты [22, 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 [24], и приводим обоснование выбора решений, основываясь на соответствующих метриках.

2.1 Постановка задачи и описание требований

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.

Рассмотрим теперь то, насколько актуальна данная проблема и какое место занимает предлагаемое решение среди существующих систем анализа групп в социальных сетях.

На рынке уже существует большое множество сервисов дающих возможность проанализировать группу на предмет возможного размещения рекламного поста [21]. Однако несмотря на такое разнообразие сервисов, их функционал слабо отличается друг от друга.

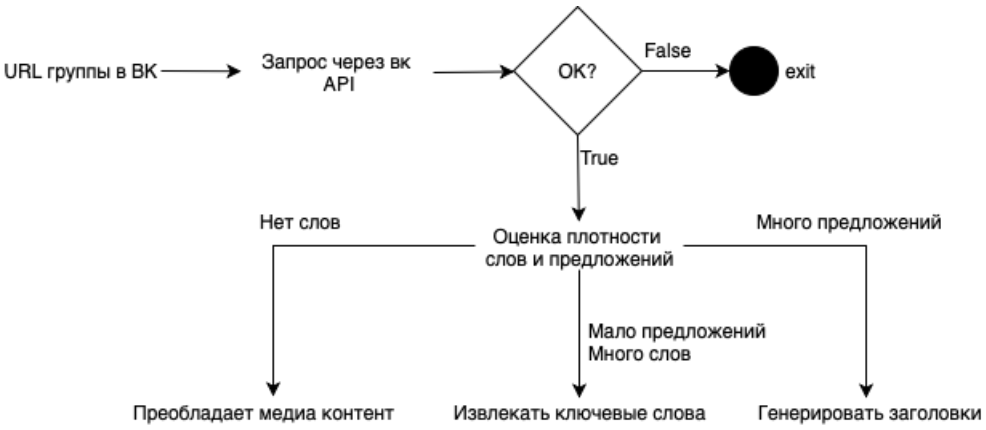

Рис. 1: Принцип работы системы.

Как правило, предоставляется возможность следить за аудиторией (ее полом, численностью, возрастом), частотой выпускаемого материала, частотой выпуска материала и его типом. Однако никто из игроков не производит анализа содержания групп, что кажется упущением возможности выделиться на рынке.

Наличие анализа содержания группы могло бы повысить скорость создания рекламы за счет уменьшения времени, необходимого для того, чтобы вникнуть в идею группы.

Причинами того, что создатели подобных сервисов не спешат с созданием таких услуг мы видим следующее:

- Необходимо нанимать новых сотрудников, что влечет дополнительные расходы;
- Спрос на подобный функционал может быть низок из-за того, что в случае успеха подобного функционала, может снизиться спрос на специалистов в области SMM.
- Поскольку спрос низкий, то данное предложение не предлагается.

Имея ввиду причины отсутствия подобного функционала, мы считаем, что есть смысл создать его в виде черной коробки, которую можно будет подключать к уже существующим сервисам. В этой черной коробке использовать алгоритмы, которые показали отличные результаты на академических датасетах и рекомендовать их как базовые, однако оставить выбор за разработчиками при выборе алгоритмов.

Таким образом стоят следующие задачи:

- Проанализировать существующие алгоритмы выделения ключевой информации из текстов;
- Предоставить возможность воспользоваться этими решениями как черным ящиком, создав интерфейсы для разработчиков:
  - Библиотеку на python
  - REST API, чтобы была возможность использовать из других языков программирования

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 [7]. Ими предлагается применять нейронные сети с архитектурой Transformer и предобработкой Byte pair encoding (BPE) [20]. 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 [16] 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 [13].

3 Описание системы

4 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.

4.1 Суммаризация текста

Для суммаризации текста мы воспользовались алгоритмом экстрактивной суммаризации основанном на TextRank [23, 17, 13], и моделью трансформера [3], обученной на датасете РИА новостей [7]. Для предобработки данных модели трансформера мы использовали byte pair encoding [20]. 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 [7], который трудно побить как экстрактивной, так и абстрактивной суммаризацией.

4.1.1 Baseline

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

4.1.2 TextRank

TextRank является является адаптацией идеи алгоритма PageRank [15] 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.

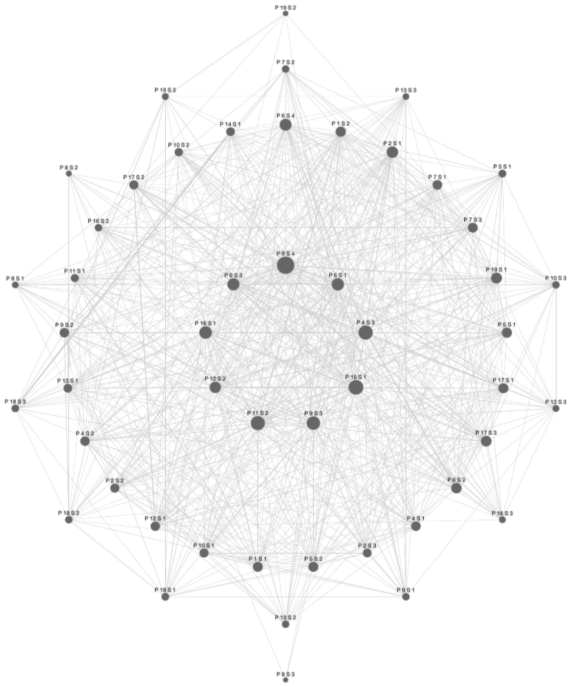

Рис. 2: Пример результирующего графа textrank.

4.1.3 Byte pair encoding

Мы используем byte pair encoding (BPE), технику, предложенную Сеннрич для задачи машинного перевода в [20]. 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.

4.1.4 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.

4.2 Выделение ключевых слов

Статья Boudin, Florian [4] рассматривает наиболее сильные подходы к выделению ключевых слов. И основная суть их статьи в том, что они предлагают реализации основных алгоритмов выделения ключевых слов, однако проблема, с которой мы столкнулись состоит в том, что в ядре их библиотеки заложена библиотека spacy [9], которая не работает с русским языком

Таким образом, для данной работы мы реализовали TopicRank, TfIdf и YAKE поскольку эти алгоритмы являются лучшими алгоритмами для выделения ключевых слов. Мы выложили эти реализации в качестве библиотеки на python [1], реализовав интерфейсы библиотеки Scikit-Learn [19, 1].

В секциях ниже мы опишем детали наших реализаций данных алгоритмов.

4.2.1 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

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

важностью в документе, и ключевые фразы извлекаются путем выбора одного кандидата ключевой фразы для каждой из наиболее важных тем.

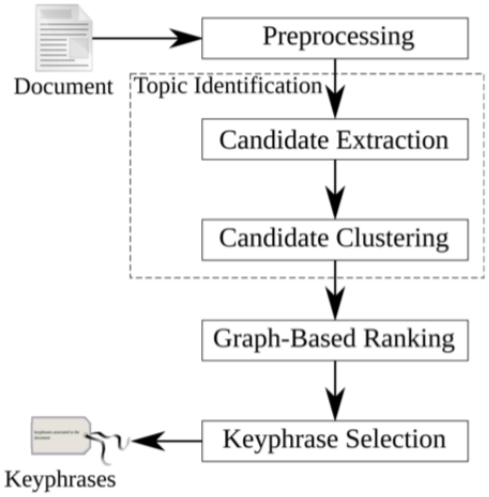

Рис. 3: Принцип работы TopicRank.

4.2.2 YAKE

4.3 Оценки качества

Для оценки качества текстовых моделей мы использовали метрику ROUGE-L F1 [11], при этом мы считали ее на датасете РИА новостей [7]. Помимо этого, на основе этого датасета проводилось соревнование по генерации заголовков, где автором было получено 3 место, а описание результатов соревнования было подано в качестве статьи на конференцию "Диалог".

5 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 [12], недавно опубликованным датасетом Риа новостей [7], а также в нашей работе мы используем данные из 25 групп вконтакте с разным целевым материалом: от медиа контента до полноценных длинных текстов.

5.1 Данные выделения ключевых слов

Для анализа качества использованных алгоритмов для выделения ключевых слов мы воспользовались датасетом с разнообразными статьями на русском языке [12]. В наборе присутствуют научные статьи из "журнала киберленика"[5], технического блога "хабрахабр"[8] и новостных ресурсов "Независимая газета"[10] и "Россия сегодня"[18].

Датасет	**Тип**	**Количество документов**	**Среднее количество токенов**
Киберленика	Абстракты	4072	5.27
Habr	Блоги	3990	5.03
Независимая газета	Новости	1987	6.11
Россия сегодня	Новости	7217	10.07

5.2 Данные для генерации заголовков

Для обучения и выбора алгоритмов генерации заголовков, мы воспользовались недавно опубликованным датасетом Риа новостей. Этот датасет содержит новости с января 2010 по декабрь 2014. В нем имеется 1003869 новостных статей со средним размером заголовка 9.5 слов и средней длинной текста 315.6 слов.

Эти датасет предоставлен в виде необработанных фрагментов оригинальных html-страниц. Это означает, что в данных присутствовали различные HTML-теги и объекты. В итоге имеется необработанная новость и соответствующий заголовок.

```
<p> <strong> <\strong> <\p> \n <p> <strong> Moscow,
Dec 1   &mdash; RIA news. <\strong> a fire in  
one of the   workshops in  ...<\p>
```

В результате первым делом мы попытались очистить данные от ненужной информации. И так, мы создали препроцессор, который удаляет все html-теги и сущности.

Кроме того, мы обнаружили, что иногда в данных отсутствует текст, поскольку исходные новости представлены в виде изображений (например, снимок экрана Twitter), а новости чисто польские. Это все выбросы, с которых мы очистили данные.

6 Эксперименты

В данной секции мы приводим технические детали, параметры моделей и оценки качества созданных моделей на датасетах описанных выше.

6.1 Генерация заголовка

На практике обычно используется метрика ROUGE [**?**] при оценке качества алгоритмов суммаризации. Конкретно используются так называемые ROUGE 1,2, L - точность, полнота и F1. Имена в ROUGE X - Y обозначают следующее: X - количество n-грамм, используемых для вычисления метрики Y. В случае n-грамм размера L рассматривается самая длинная общая подпоследовательность из предсказанной последовательности найденая в исходной последовательности. Метрики Y - классическая точность, полнота и их среднее гармоническое.

В случае алгоритма Textrank [23], алгоритма экстрактивной суммаризации, мы воспользовались реализацией из библиотеки gensim [17]. В случае этого алгоритма, выделяется не одно предложение, а подмножество текста определенного размера, потому мы установли параметр отвечающий за размер возвращаемого текста равным 20% от исходного.

Мы взяли state of the art реализацию алгоритма Universal Transformer из Open NMT [14] и обучили этот алгоритм на данных Риа новостей. Мы использовали 4 слоя кодировщика и декодеривщика, 8 heads of attention, вероятность дропаута была выставлена равной 0.3. Для оптимизации был использован алгоритм Adam с изменяющейся скоростью обучения, по правилу из оригинальной статьи о трансформере.

В качестве входа модели мы использовали первые 2000 BPE токенов.

Что касается обучения Byte Pair Encoder, то мы попробовали пердобученный на википедии токенизатор и обученный на датасете Риа новостей. В таблице 1 они обозначены Wiki BPE и Ria BPE, соответственно.

Также для отбора лучших кандидатов для заголовка мы использовали beam search размером 10.

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	0.37	0.39	0.36	0.20	0.21	0.19	0.34	0.37	0.34
Ria BPE transformer	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

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

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 2: Лучшие и худшие результаты предсказания модели. Score среднее от ROUGE-1,2,L F1.

В Таблице 2 среднее от ROUGE-1,2, L F1 используется в качестве метрики. Мы показываем два примера: лучшее и худшее возможное предсказание нашего лучшего алгоритма – Wiki transformer-a.

6.2 Выделение ключевых слов

Как видно из Таблицы 3, по метрике F1, алгоритмы TopicRank и YAKE не ступают друг-другу, при этом YAKE показывает высокую точность, а TopicRank обладает высокой полнотой.

Так что для конкретных задач есть смысл выбирать конкретный алгоритм: Если требуется скорость, то хороший выбор TfIdf, поскольку в нем нет каких либо трюков с признаками, а просто считается статистика. Если важно, чтобы было как можно меньше неправильных ключевых слов, то стоит использовать YAKE, а если важно то, чтобы извлекалось как можно больше релевантных примеров, то следует использовать TopicRank.

Алгоритм/Датасет	P	R	F
TfIdf	13	24	16
TopicRank	16	24	19.2
YAKE	23.5	16.6	19.3

Таблица 3: Средняя абсолютная точность (P), полнота (R), и F1 мера на всех датасетах

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. Также 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] Boudin, F. pke: an open source python-based keyphrase extraction toolkit / Florian Boudin // Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations. — Osaka, Japan, 2016. — December. — P. 69–73. — http://aclweb.org/anthology/C16-2015.

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

[6] 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.

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

[8] Habr. — https://habr.com/ru/.

[9] Honnibal, M. An improved non-monotonic transition system for dependency parsing / Matthew Honnibal, Mark Johnson // Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. — Lisbon, Portugal: Association for Computational Linguistics, 2015. — September. — P. 1373–1378. — https://aclweb.org/anthology/D/D15/D15-1162.

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

[11] 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.

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

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

[14] OpenNMT: Open-Source Toolkit for Neural Machine Translation / G. Klein, Y. Kim, Y. Deng et al. // ArXiv e-prints.

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

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

[17] 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.

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

[19] 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.

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

[21] smm pub. 20 websites for vk groups analysis. — 2019. — may. — https://vk.com/page-43503600_49056860.

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

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

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