Инженерный вестник Дона, №10 (2020)
ivdon.ru/ru/magazine/archive/n10y2020/6651

Обзор задачи автоматической суммаризации текста

А.Ю. Белякова[1], *Ю.Д. Беляков*[2]

[1]*Иркутский государственный аграрный университет имени А.А.Ежевского, Иркутск*
[2]*Санкт-Петербургский государственный университет, Санкт-Петербург*

Аннотация: Статья посвящена обзору задачи суммаризации текста в целом и анализу различных способов решения данной проблемы. С этой целью было выполнено сравнение нескольких методов суммаризации с описанием используемых в сравнении инструментов, проведен анализ качества и скорости алгоритмов, относящихся к различным типам, представлены результаты выполненного анализа, а также представлен пример работы. Обзорное исследование может помочь тем, кто хочет получить базовое понимание задачи суммаризации или области NLP.
Ключевые слова: методы суммаризации, машинное обучение, обработка естественного языка.

Введение

Обработка естественного языка (Natural Language Processing, NLP) - общее направление искусственного интеллекта и математической лингвистики, в нем изучаются проблемы компьютерного анализа и синтеза естественных языков. Задачи NLP направлены на создание таких систем, которые способны выполнять задачи, связанные с языком, на таком же уровне как и человек. Одна из таких задач - автоматическая суммаризация текста (Automatic Text Summarization, ATS).

Краткое содержание любого документа имеет невероятную ценность. Оно сохраняет нам время, позволяет оценить релевантность текста к предмету нашего поиска, с его помощью мы способны презентовать любую текстовую информацию, не вдаваясь в подробности, и т.д. Создание краткого содержания вручную - это трудоемкий и времязатратный процесс, требующий знания полного содержания документа и способности выделить ключевые аспекты [1, 2]. Изложения могут разниться от одного человека к другому. С ростом объема информации необходимость исследований в этой области только увеличивалась. Мы можем получить нужные нам знания как угодно и где угодно, однако избыточные данные могут затруднять и

замедлять нам поиск. Таким образом, изучение автоматических способов отделения полезной информации от всего остального имеет очевидные преимущества.

Автоматическое создание таких ценных отрывков текста - давно известная проблема в области обработки естественного языка (Natural Language Processing). Работа Ханса Питера Луна [3], посвященная решению этой задачи, была одной из первых, с тех пор появилось огромное количество различных методов решения.

Классификация методов суммаризации

Выделяется множество групп, на которые делятся все методы суммаризации. Подходы могут отличаться по формату вывода, по формату ввода, по подробностям, по содержанию вывода и по назначениям. Рассмотрим все группы по порядку.

По формату вывода методы делятся на экстрактные (extractive) и абстрактные (abstractive).

1. Экстрактные методы алгоритмически находят подмножество самых информативных частей текста, как правило, предложений, и составляют из них краткое содержание.

2. Абстрактные методы генерируют новый текст краткими фразами, которые семантически согласуются с изначальным документом и содержат его важнейшую информацию. Результаты работы таких решений похожи на то, как люди пересказывают тексты, но такие решения сложнее реализовать.

По формату ввода методы делятся на *Single Document* и *Multi Document*.

1. *Single Document* подходы принимают на вход лишь один документ(текст). Их минус следует из названия: такой метод не может обработать несколько документов с похожей темой.

2. *Multi Document* подходы способны принимать на вход несколько документов и суммаризовать их, однако такие алгоритмы сложны в имплементации.

По подробностям методы делятся на ориентировочные и информативные.

1. Ориентированные подходы помогают пользователю лишь ознакомиться с идеей текста, они используются для категоризации документов. Данный алгоритм легко имплементируем.

2. Информативные методы дают краткий обзор всего текста. Продукт работы такого алгоритма можно использовать как замену тексту.

По содержанию вывода методы делятся на генеральные и основанные на запросе.

1. Результат работы генеральных методов не зависит от пожеланий пользователя. Краткое содержание, сгенерированное таким алгоритмом, передает темы и идеи текста, которые оцениваются алгоритмом больше других.

2. Методы основанные по запросу выдают результат, который направлен на покрытие запроса пользователя. С помощью таких подходов можно найти желаемую информации в тексте, игнорируя все ненужное.

По назначению методы делятся на направленные и ненаправленные на тему.

1. Направленные методы имеют зафиксированную тему, и подходят только для суммаризации документов, относящихся к такой теме.

2. Ненаправленные подходы не привязаны к какой-то отдельной области или теме. Они подходят для любого текста и в среднем будут показывать результаты, лучшие относительно направленных методов, за исключением того топика, на который они направлены.

Способы подготовки текста

Прежде чем начать обрабатывать текст и пытаться его сократить, к начальному тексту чаще всего применяют некоторый перечень из способов предобработки.

Предобработка текста необходима для приведения текста к такому виду, чтобы результат работы алгоритма был как можно лучше. Большинство методов предобработки направлены на приведение составляющих текста к единому виду и его очистке. Основные способы подготовки текста:

1. Перевод всех букв в нижний регистр.
2. Удаление или конвертация в словах всех чисел, дат и т.д.
3. Удаление пунктуации.
4. Расшифровка аббревиатур.
5. Удаление стоп-слов (бесполезных слов с точки зрения смысловой нагрузки, например: союзы, артикли и т.д.)
6. Стемминг (процесс приведения однокоренных слов к общему виду, посредством удаления изменяемых частей слова).
7. Лемматизация (процесс приведения однокоренных слов к общему виду посредством изменения слова так, чтобы оно приняло свою начальную форму).
8. Токенизация (процесс преобразования текста в список предложений, а предложений в список слов). Это необходимо для того, чтобы работать с текстом как с последовательностью объектов, а не как со строкой.
9. Разметка частей речи. Существует большое количество инструментов, позволяющих определить часть речи слова, его роль в предложении и т.д. Такая информация используется в алгоритмах как

дополнительный критерий отбора. Например, в подлежащих и глаголах содержится больше ценной информации, нежели в союзах и предлогах.

10. Распознавание имен собственных.

11. Разрешение ссылок. Часто в тексте используются местоимения и другие способы “ссылки” на некоторое слово с целью избегания повторений. Если такие части текста мешают нормальному функционированию алгоритма, с помощью *coreference resolution* можно заменить их на те слова, на которые они ссылаются.

Это основные и чаще всего используемые методы предобработки, но нередко используются и другие, пользовательские, подходы. Мы рассмотрим статистические, графовые, лингвистические подходы, а также те, которые включают в себя использование машинного обучения.

Статистические - одни из первых подходов, которые применялись в задаче суммаризации. Они основаны на оценивании всех предложений по некоторому признаку. После оценивания, краткое содержание составляется из предложений, имеющих наибольшие значения этих признаков. Примерами таких оценочных параметров могут быть:

- частота слов;
- положение предложения в тексте;
- длина предложения;
- близость предложения к заголовку или подзаголовку.

Нередко используют комбинации таких признаков, однако это не гарантирует прироста качества.

Плюсы статистического подхода.

1. Нетребовательны к ресурсам.
2. Простые и быстрые.

Минусом данного метода является читабельность. Такие методы могут “вырывать” предложения из контекста, в результате отрывки текста могут не

иметь смысла для читающего. Эта проблема может быть решена, например, добавлением дополнительных фильтров на отбор предложений.

В качестве примера статистического подхода рассмотрим метод TF-IDF.

TF-IDF [4] - это экстрактный статистический метод, основанный на одноименной метрике. TF-IDF (term frequency-inverse document frequency) - это численный признак слова, показывающий как часто слово встречается в предложении и в множестве предложений. TF-IDF рассчитывается по следующей формуле:

tfidf(w) = *tf(w)***idf(w)*, где *tf(w)* = (Сколько раз слово встретилось в предложении) / (Число слов в предложении), *idf(w)* = *log*(Количество предложений / Количество предложений со словом w в них)

Чтобы посчитать TF-IDF для всего предложения, достаточно сложить значения TF-IDF слов в этом предложении.

Алгоритм суммаризации, основанный на этом методе, будет выглядеть следующим образом:

1. загрузить текстовый документ;
2. предобработать текст;
3. посчитать TF-IDF значения для каждого слова;
4. посчитать TF-IDF значения для каждого предложения;
5. составить краткое содержание.

Графовые подходы основаны на представлении предложений текста в виде графа, используя меру сходства, а затем эта структура используется для подсчета важности каждого предложения.

В качестве примера такого подхода рассмотрим TextRank.

TextRank [5] - графовый алгоритм, главной идеей которого является “рекомендация”. Каждое предложение выполняет роль вершины направленного графа. Если вершина *A* соединяется с вершиной *B*, это значит

что *B* получила “голос” или “рекомендацию” от *A*. Чем больше число голосов у вершины, тем больше важность предложения, более того, важность вершины, которая отправляет голос, влияет на важность голоса.

Рекомендацию можно определить по-разному. Для задачи суммаризации такой “голос” определяют, как сходство двух предложений. В оригинальном предложенном алгоритме, сходство предложений вычисляется как нормализованное пересечение этих предложений.

Сходство $(Si,Sj) = |\{wk \mid wk \in Si \& wk \in Sj\}| / (\log(|Si|)+\log(|Sj|))$,

где *Si* - i-ое предложение, состоящее из слов *{wk}*.

Алгоритм суммаризации основанный на этом методе будет выглядеть следующим образом:

1. загрузить текстовый документ;
2. предобработать текст;
3. составить граф, вершинами которого будут являться предложения;
4. получить оценки сходства для каждой пары предложений;
5. отсортировать все вершины и составить краткое содержание.

В последнее время большое внимание уделяется моделям глубокого обучения в различных задачах NLP. Именно такие модели показывают State-of-the-Art результаты, в том числе для задачи суммаризации текста. Эти методы основаны на больших нейронных сетях, которым требуется огромное количество данных для тренировки.

Характерная особенность методов, основанных на глубоком обучении - краткое содержание, похожее семантически на составленное человеком. Однако очень сложно добиться такого результата для любого входного текста. Большое количество параметров, необходимость огромного количества данных для обучения, валидации и тестов - все это влияет на

сложность в их имплементации, их время работы и сложность вычислений. Примером такой модели является BART.

BART [6] - это шумоподавляющий автоэнкодер, который был натренирован на задании восстановления поврежденных текстов. Предобученные версии этой модели показывают отличные результаты на *seq2seq* заданиях. *Sequence-to-sequence* - это задачи, в которых на вход подается некоторая последовательность (в нашем случае это последовательность слов, то есть предложение), а на выходе последовательность в другом виде. К таким задачам относятся: перевод текста, суммаризация, генерация текста, и др.

Сравнение и результаты

Мы будем сравнивать несколько методов, относящихся к различным группам. Оценочными критериями будут являться ROUGE-оценка и время исполнения. Сравнение будет проходить следующим образом: на одном для всех наборе данных будут измеряться оценки качества, затраченная вычислительная мощность, время работы.

В качестве данных для проверки будет использоваться датасет CNN (Cable News Network).

CNN содержит новостные статьи и выдержки из них. Эти данные были собраны с веб-страниц CNN. Из-за ограничения в вычислительной мощности компьютера, на котором проводилось сравнение, эвалюация методов проводилась не на полном наборе данных CNN (92579 статей и кратких содержаний), а лишь на их фракции (10000 статей и кратких содержаний).

Сравниваться будут следующие методы:

1. TextRank.
2. Luhn.
3. Latent Semantic Analysis, LSA.
4. LexRank.

5. SumBasic.

6. KL-Sum.

Метод TextRank был описан выше. Рассмотрим все остальные подходы.

LSA [7] это, как уже было сказано, алгебраический метод. Этот алгоритм извлекает скрытые семантические структуры слов и предложений с помощью сингулярного разложения.

Алгоритм состоит из 3 шагов:

1. Создание матрицы: Входной текст представляется в виде матрицы, чтобы было возможно производить над ним вычисления. Размерность матрицы – (*Количество слов) x (Количество предложений)*. Таким образом, каждый элемент матрицы должен отвечать за важность слова в предложении. Например, заполнить матрицу можно метриками TF или TF-IDF.

2. Полученная на первом шаге матрица сингулярно раскладывается. Это делается с целью получить представление матрицы, как векторов в Евклидовом пространстве. Помимо того, что мы получаем способность моделировать взаимоотношения слов и предложений, сингулярное разложение уменьшает шум, а это увеличивает точность алгоритма.

3. Выбор предложений. Пусть *dj* - вектор, соответствующий документу *j*, показывающий ценность каждого слова в этом документе, а *tiT*-вектор, соответствующий слову *i*, показывающий ценность этого слова в каждом предложении. Тогда сингулярное разложение матрицы выглядит следующим образом (см. Рис. 1).

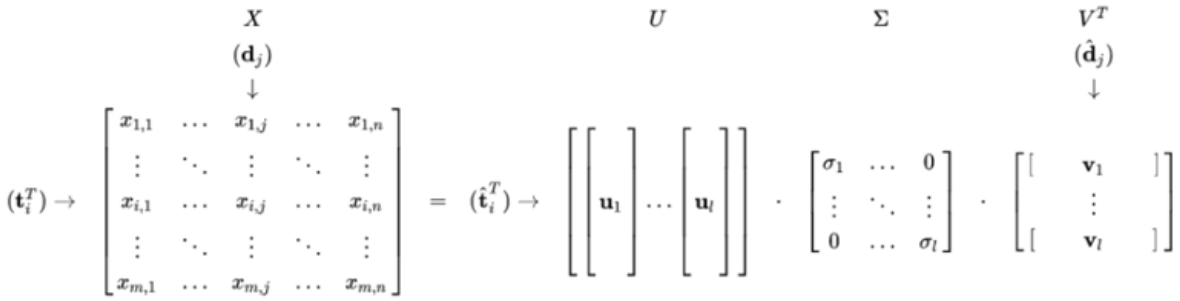

Рис. 1. - Сингулярное разложение матрицы

Из трех векторов, полученных в результате разложения, именно матрица *VT* несет в себе информацию о силе каждого слова в предложении. Использование этих матриц разнится в зависимости от вариации LSA метода. В сравнении мы будет использовать подход, предложенный в [8].

Luhn. Первый известный метод решения задачи суммаризации. Основан на предположении о том, что самые значимые предложения - те, которые содержат наибольшее количество значимых слов (значимые слова - самые часто встречающиеся не стоп-слова).

Алгоритм сводится к трем этапам:

1. Создание матрицы TF-IDF для каждого слова (после исключения стоп-слов) и предложения в тексте. Эта операция уже была описана ранее.

2. Оценка предложений рассчитывается по следующей формуле:

$$Оценка = (ЧИСЛО\ ЗНАЧИМЫХ\ СЛОВ\ В\ ПРЕДЛОЖЕНИИ)2ЧИСЛО\ ЗНАЧИМЫХ\ СЛОВ\ В\ ТЕКСТЕ,$$

где значимые слова - слова имеющие наибольшие TF-IDF оценки.

3. Выбор предложений с наилучшими оценками.

LexRank [9] Графовый метод, основанный на TextRank. В отличие от TextRank, этот подход использует другой метод оценивания схожести двух предложений для составления весов графа. Все предложения представляются в виду TF-IDF векторов, а сама оценка схожести вычисляется так:

similarity = *A B*||*A*|| ||*B*||, где *A* и *B* векторные TF-IDF представления предложений. Такая мера сходства называется Cosine Similarity.

SumBasic [10] Статистический алгоритм, основанный на идее о том, что относительная частота употребления набора не стоп-слов в тексте - это хороший способ предсказать появление того или иного слова в кратком содержании, сгенерированном человеком. В таком методе предложения оцениваются следующим образом:

$Score(S) = w \in S1|S|PD(w)$, где $PD(w)$ вероятность появления слова wполученная из документа D. Краткое содержание составляется из предложений, набравших наибольшую оценку.

KL-Sum [11, 12]. В данном алгоритме для выбора краткого содержания S используется следующий критерий:

$S = min\ KL(PD||PS)$, где $KL(PD||PS)$ это расстояние Кульбака - Лейблера между распределениями вероятности появления слов в изначальном документе и в сгенерированном кратком содержании.

Критерии оценивания.

Все методы будут оцениваться по метрике ROUGE. ROUGE - это оценка, которая активно используется для оценивания качества моделей в задачах суммаризации и машинного перевода. Ее идея очень проста: ROUGE-оценка показывает, как сильно пересекаются два разных текста. В нашем случае два текста это “правильное” краткое содержание и сгенерированное.

Формально, ROUGE-n - это гармоническое среднее между ROUGE-n-precision и ROUGE-n-recall, где

ROUGE-n-precision = КОЛИЧЕСТВО ПЕРЕСЕКАЮЩИХСЯ СЛОВОСОЧЕТАНИЙ ИЗ n СЛОВ КОЛИЧЕСТВО СЛОВОСОЧЕТАНИЙ ИЗ n СЛОВ В СГЕНЕРИРОВАННОМ ТЕКСТЕ,

ROUGE-n-recall = КОЛИЧЕСТВО ПЕРЕСЕКАЮЩИХСЯ СЛОВОСОЧЕТАНИЙ ИЗ n СЛОВ КОЛИЧЕСТВО СЛОВОСОЧЕТАНИЙ ИЗ n СЛОВ В ПРАВИЛЬНОМ ТЕКСТЕ.

В итоговой таблице будут использованы 3 версии ROUGE-оценки: ROUGE-1, ROUGE-2 и ROUGE-L, где *L* - означает, что поиск пересечений идет не по фиксированному размеру словосочетаний, а по наибольшему.

Применяемая предобработка

Для всех методов была применена одна и та же предобработка: удаление всех небуквенных символов, перевод в нижний регистр, удаление стоп-слов и стемминг.

Результаты

В таблице 1 отображены результаты сравнения шести алгоритмов на наборе данных CNN. Можно заметить, что лучший результат по ROUGE-оценке показал метод LexRank с оценками ROUGE-1 = 0.2531, ROUGE-2 = 0.0867, ROUGE-L = 0.1833. Это означает, что среди этих шести алгоритмов LexRank будет показывать наилучшие результаты на документах, похожих на используемый, для сравнения набора данных CNN.

Таблица № 1

Сравнение методов суммаризации по ROUGE-оценкам

Метод\Оценка	ROUGE-1	**ROUGE-2**	**ROUGE-L**
TextRank	0.2343	0.0833	0.1722
LSA	0.2121	0.0705	0.1515
Luhn	0.2397	0.0861	0.1752
LexRank	0.2531	**0.0867**	**0.1833**
KL-Sum	0.1936	0.0658	0.1430
SumBasic	0.2463	0.0697	0.1755

В таблице 2 показан результат сравнения времени выполнения алгоритмов. Время в таблице - это время, за которое каждый алгоритм обработал и суммаризировал 100 документов. Лучшие результаты показали SumBasic, Luhn и LexRank.

Таблица № 2

Сравнение методов суммаризации по времени выполнения

Метод	**TextRank**	**LSA**	**Luhn**	**LexRank**	**KL-Sum**	**SumBasic**
Время	4,35 сек.	6,77 сек.	2,33 сек.	2,85 сек.	11,2 сек.	1,58 сек.

Пример работы

Изначальный текст:

A U.S. envoy expressed optimism that food aid would find its way to those in need in North Korea after two days of talks with officials from Pyongyang, but it remained unclear when the shipments might begin. Robert King, the U.S. special envoy for North Korean human rights issues, met with North Korean representatives in Beijing to thrash out the details of a plan to allow the resumption of food aid to the North. «We resolved the administrative issues that we were concerned with, «King said Thursday before leaving for Washington to report the results of the discussions. He described the meetings as «very productive, positive talks. «He added, though, that the timing of the food deliveries was not yet clear. «We're still working on the details, " he said. » Not all of those questions have been worked out. "North Korea last week announced an agreement to freeze its nuclear and missile tests, along with uranium enrichment programs, and allow the return of U.N. nuclear inspectors. The United States said it would provide 240,000 metric tons of nutritional assistance to the impoverished country. The United States had suspended shipments of food aid to North Korea in 2009 amid tensions over Pyongyang's nuclear program and concerns that the supplies were not reaching those most in need. The initial deal last week to resume the deliveries came after the two countries revived negotiations that had stalled after the death in December of the longtime North Korean leader Kim Jong Il. The talks this week were held in order to finalize points like what ports will be used to dock incoming ships, how the distribution of the food will be monitored and which

nongovernmental organizations will be involved. The agreement last week was cautiously welcomed by U.S. officials in the hope that a new era in relations with the North would begin and lead to a resumption of multilateral talks aimed at the denuclearization of the Korean peninsula. But Pyongyang has stepped up its rhetoric against the South Korean president, Lee Myung-bak, and his government since Kim Jong Un took over from his father, Kim Jong Il, as North Korean leader. Earlier this week, North Korean television aired footage of a military unit carrying out live-fire drills in sight of a South Korean island. It showed tanks repositioning and an artillery machine being prepared, overlooking waters that have seen a number of violent incidents over the years. North Korea shelled Yeonpyeong Island in November 2010, killing four South Koreans, claiming it was responding to a South Korean military drill in the area. Li Gum-chol, a North Korean deputy commander, said: «We will turn Seoul into a sea of flames by our strong and cruel artillery firepower, which cannot be compared to our artillery shelling on Yeonpyeong Island. We are training hard, concentrating on revenge to shock Lee Myung-bak's traitorous group and the military warmongers in South Korea. "The United States and South Korea are carrying out annual joint military drills, which North Korea has condemned as a provocation. Now, Pyongyang is staging its own. CNN's Chi-Chi Zhang and Jethro Mullen contributed to this report.

Краткое содержание, сгенерированное алгоритмом LexRank:

Robert King, the U.S. special envoy for North Korean human rights issues, met with North Korean representatives in Beijing to thrash out the details of a plan to allow the resumption of food aid to the North. " The United States had suspended shipments of food aid to North Korea in 2009 amid tensions over Pyongyang's nuclear program and concerns that the supplies were not reaching those most in need. North Korea shelled Yeonpyeong Island in November 2010, killing four South Koreans, claiming it was responding to a South Korean military

drill in the area. "The United States and South Korea are carrying out annual joint military drills, which North Korea has condemned as a provocation.

Заключение

Задача суммаризации текста остается актуальной, её важность растёт, и заставляет исследователей NLP разрабатывать и предлагать новые улучшенные варианты её решения. В данной статье рассмотрена классификация методов суммаризации, выполнен сравнительный анализ качества и скорости алгоритмов, относящихся к различным типам, представлены результаты анализа, приведен пример работы.

В общем, перед использованием какого-либо метода решения задачи ATS необходимо определить, подходит ли он в рассматриваемом случае. Методы, показывающие себя хорошо в случае новостных статей, могут быть плохи для литературных произведений. Если важна скорость алгоритма, то лучше обращаться к статистическим методам, если качество результата - то модели глубокого обучения будут лучшим вариантом, графовые методы предоставят баланс между скоростью и качеством. Стоит учитывать, что любой из предлагаемых методов предоставляет лишь некий шаблон краткого содержания, который, предполагаемо, содержит самую важную информация текста. После составления краткого содержания можно применить к нему другие методы NLP, которые помогут избавиться от нечитаемости текста, от проблем со связностью предложений и др. Вне зависимости от поставленной задачи всегда можно найти тот метод, ту предобработку и ту коррекцию результата, которыми она будет решена.

Список литературы

1. Иванова Д.Н., Яровая Л.Е. Модели анализа словообразования в современном английском языке // Инженерный вестник Дона. 2020. №8. URL: ivdon.ru/uploads/article/pdf/IVD_69__6_Ivanova.pdf_2cabba7b2d.pdf.

2. Усталов Д.А., Берсенёв А.Ю., Киселёв Ю.А. Автоматизация процесса коллективного построения лингвистических ресурсов // Инженерный вестник Дона. 2018. №1. С. URL: ivdon.ru/uploads/article/pdf/IVD_191_ustalov_bersenev_kiselev.pdf_cdfa064934.pdf.

3. Luhn H. P. The Automatic Creation of Literature Abstracts // IBM Journal of Research and Development. 1958. №2. pp. 159-165.

4. Christian, Hans., Agus, Mikhael, Suhartono, Derwin Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF-IDF) // ComTech: Computer, Mathematics and Engineering Applications. 2016. №4. pp. 285-294.

5. Mihalcea R., Tarau P. TextRank: Bringing Order into Text // Department of Computer Science. University of North Texas. 2004. pp. 404-411.

6. Lewis M., Liu Y., Goyal, N., Ghazvininejad M., Mohamed A., Levy O., Stoyanov V., Zettlemoyer L. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. // arXiv preprint arXiv:1910.13461. 2019.

7. Ozsoy M., Alpaslan F., Cicekli I. Text summarization using Latent Semantic Analysis // All content following this page was uploaded by Ferda Nur Alpaslan. 2014. №18. pp. 405-417.

8. Steinberger J., Jezek K. Using Latent Semantic Analysis in Text Summarization and Summary Evaluation // Proceedings of the 7th International Conference ISIM, 2004. 8p.

9. Erkan G., Radev D.R. Lex Rank: Graph-based Lexical Centrality as Salience in Text Summarization. Journal of Artificial Intelligence Research // Journal of Artificial Intelligence Research. 2004. №22. pp. 457-479.

10. Aria H., Lucy V. Exploring content models for multi-document summarization. // Proceedings of Human Language Technologies, Annual Conference of the Noth American Chapter of the Association for Computational Linguistics. 2009. pp. 362-370.

11. Nenkova A., Wandervende L. The impact of frequency on summarization // Technical report, Microsoft Research. 2005. URL: researchgate.net/publication/228878974_The_impact_of_frequency_on_summarization

12. Aries A., Zegour D., Walid H. Automatic text summarization: What has been done and what has to be done // arXiv:1904.00688. 2019. 2019. pp. 1-34.

References

1. Ivanova D.N. Jarovaja L.E. Inzhenernyj vestnik Dona. 2020. №8. URL: ivdon.ru/uploads/article/pdf/IVD_69__6_Ivanova.pdf_2cabba7b2d.pdf.

2. Ustalov D.A., Bersenjov A.Ju., Kiseljov Ju.A. Inzhenernyj vestnik Dona. 2018. №1. URL:ivdon.ru/uploads/article/pdf/IVD_191_ustalov_bersenev_kiselev.pdf_cdfa064934.pdf.

3. Luhn H. P. IBM Journal of Research and Development. 1958. №2. pp. 159-165.

4. Christian H., Agus M., Suhartono D. ComTech: Computer, Mathematics and Engineering Applications. 2016. №4. pp. 285-294.

5. Mihalcea R., Tarau P. Department of Computer Science. University of North Texas. 2004. pp. 404-411.

6. Lewis M., Liu Y., Goyal, N., Ghazvininejad M., Mohamed A., Levy O., Stoyanov V., Zettlemoyer L. arXiv preprint arXiv:1910.13461. 2019.p

7. Ozsoy M., Alpaslan F., Cicekli I. All content following this page was uploaded by Ferda Nur Alpaslan. 2014. №18. pp. 405-417.

8. Steinberger J., Jezek K. Proceedings of the 7th International Conference ISIM, 2004. 8p.

9. Erkan G., Radev D.R. Journal of Artificial Intelligence Research. 2004. №22. pp. 457-479.

10. Aria H., Lucy V. Proceedings of Human Language Technologies, Annual Conference of the Noth American Chapter of the Association for Computational Linguistics. 2009. pp. 362-370.

11. Nenkova A., Wandervende L. Technical report, Microsoft Research. 2005.URL:researchgate.net/publication/228878974_The_impact_of_frequency_on_summarization

12. Aries A., Zegour D., Walid H. arXiv:1904.00688. 2019. 2019. pp. 1-34.