Министерство образования и науки Российской Федерации
Московский физико-технический институт (государственный университет)
Факультет управления и прикладной математики
Кафедра «Интеллектуальные системы»
при Федеральном исследовательском центре
«Информатика и управление» РАН

Власов Андрей Валерьевич

Методы полуавтоматической суммаризации подборок научных статей

03.04.01 — Прикладные математика и физика

Выпускная квалификационная работа
(Магистерская диссертация)

Научный руководитель:
профессор РАН, д.ф.-м.н.
Воронцов Константин Вячеславович

Москва
2020 г.

Содержание

Аннотация

Реферирование текстовых документов (Multi-document Summarization) – одна из ключевых задач в NLP. Большинство исследований нацелены на полностью автоматическое реферирование заданной подборки документов. Тем не менее, на практике создание реферата высокого качества едва ли возможно без участия человека, который имеет собственные авторские представления о целях, приоритетах и сценарии реферата. Эта информация отсутствует в исходной подборке документов и потому принципиально не может быть учтена алгоритмами автоматического реферирования.

В этой ситуации автору нужен не алгоритм, который напишет реферат полностью за него, а система подсказок, предлагающая последовательность цитирования документов и варианты продолжения текста фраза за фразой. Человек решает творческие задачи, определяя структуру нарратива и оставаясь главным его автором. Машина решает вспомогательные рутинные информационно-поисковые задачи, сокращая затраты времени человека. Такая гибридная человеко-машинная технология называется автоматизированной авторской суммаризацией текстов (Machine Aided Human Summarization).

Для построения такой гибридной технологии в данной работе предлагается конвейер (pipeline) задач машинного обучения. Две основные задачи – ранжирование документов подборки в порядке цитирования и ранжирование фраз о заданном документе в порядке релевантности. Предлагается пользовательский интерфейс для написания реферата с суфлерами (prompters), основанный на этих двух задачах. Для обучения алгоритмов ранжирования предлагается строить обучающую выборку из текстов научных статей.

В каждой статье имеется список литературы (references), который является неупорядоченной подборкой документов, и обзорная часть текста, которая суммаризирует эти документы. Для формирования такой выборки конвейер был дополнен вспомогательными задачами машинного обучения. К ним относятся: выделение списка статей библиографии, выделение внутритекстовых библиографических ссылок, их связывание с пунктами библиографии, выделение обзорной части из текста статьи.

Суфлёров может быть много, и каждый их них служит для ранжирования фраз, относящихся к определённому аспекту статьи. В данной работе подобно рассматривается суфлёр, который подсказывает, как на данную статью обычно ссылаются из других статей. Другие суфлёры обучаются выделению различных аспектов из текстов научных статей: цель и позиционирование исследования, теории, материалы и методы, датасеты, результаты, выводы. Для обучения каждого такого суфлёра ставится отдельная задача машинного обучения для выделения из текста научной статьи фраз, наиболее релевантных выделяемому аспекту.

В экспериментальной части работы исследуются отдельные конструктивные блоки конвейера, описываются методики формирования данных, сравниваются различные методы. Предлагается общая методика оценивания качества гибридной технологии суммаризации, не требующая ни экспертных оценок, ни протоколирования взаимодействия пользователей с системой в процессе реферирования. Для предлагаемых методов результаты сравнения превосходят текущий уровень state-of-the-art.

1 Introduction

The published papers of the scientists double approximately every nine years according to the statistics [8]. So, this is the reason why scientists and experts have to spend a lot of time reading scientific publications and writing reviews. Therefore they become get with textual information. Sometimes a researcher does not need to go into details and read the entire paper. Hence, a brief summary of the work may help to grasp the relevance of the work. (Should he use (read) it or not?) As compared to the abstract part the brief summary highlights not only the main points according to the author's opinion but also an impact of this paper on scientific community.

A great deal of cognitive effort is required to accomplish a summary: different fragments of a text are to be selected, reformulated and assembled according to their relevance. The coherence of the information included in the summary is to be taken into account as well. [60]

Is artificial intelligence able to shorten this time? In this case the text summarization task occurs.

Definition 1.1. Automatic Text Summarization is an automatic procedure of writing a concise text from the whole one while establishing the important points. Moreover, the final summary should be a coherent text. [60]

1.1 Motivation

Frequently, a research group has to solve the task of collecting scientific papers or conducting a survey. The examples of situations where this problem may occur:

- to research novel area of knowledge;
- to write the grant request/technical report;
- to write an overview of thesis or paper;
- to prepare content for an educational course.

To make a review on plenty of papers the problem of multi-document summarization has to be identified.

Definition 1.2. Multi-document summarization is an automatic procedure for creating a summary which contains the key information from all these documents [60].

The following approaches are used to solve this task:

Definition 1.3. Extractive summarization is an automatic procedure for selecting the key phrases (word collocations or sentences) without any modifications and removing unnecessary ones. Usually the process is as follows: every elicited phrase is assigned with a score (according to it importance) and after that the most vital phrases are matched (according to this score) [60].

Definition 1.4. Abstractive summarization is an automatic procedure for generating a coherent summary. In this approach the original text is being shortened and re-phrased but the main logic of the text is to be preserved [60].

Summarization is classified as single-document or multi-document based on the number of source document both extractive or abstractive ones according to the methodology. In spite of the fact that automatic abstractive summarization is the desired goal in this field, nowadays researches mostly focus on extractive summarization as better results are achieved [9, 11, 41, 43]. Therefore in the case of long text summarization task the existing multi-document summarization methods are mostly extractive. This methods are divided into unsupervised, supervised and other ones.

Firstly, lexical and grammatical structure of papers have been applied in unsupervised works to generate the summary [5,25]. Later on, graph-based methods, linear programming and topic modelling approaches have been widely used for multi-document summarization, e.g., TextRank [43], LexRank [23], Integer Linear Programming [24] and ARWG [28].

Supervised approaches consider summarization as a sequential classification task. In one of the supervised approaches, Conditional Random Fields [57], the binary classification of sentences sequentially has been used. Recently, different neural model methods have been used in the majority of works or their ensemble to build learning model as well. Primarily, Seq2Seq model was commonly used in selecting recurrent neural networks [9, 15,35,47,68] as encoder, auto-regressive decoder [14,31,68] or non auto-regressive decoder [4,30,48] as decoder, based on pre-trained word representations [44,50]. Also, graph based neural models [66, 67] were used.

Furthermore, another extractive summarization approach, reinforcement learning, has been used to summarization task [14, 48, 65], which has been able to provide more direct optimization goals.

However, totally automatic summarization does not reflect the aim of reviewing scientific papers. Every researcher has its own purpose that machine (computer) is unaware a priori. For example, a person may need a review on his thesis or a grant request. Hence, he focuses on specific subjects (which are vital for him) during the summarization which the machine cannot guess. Therefore, the procedure of summarization of scientific papers has to be controlled by human.

In such situations a summary is needed to be edited before inserting it into the work (a paper or any other document) of a researcher. In fact, summary being generated by the machine may only encourage the human to write his/her own summary in less time. Also an opportunity to go to the source text while reading the review should be presented as it may be useful for the researcher to upgrade the machine summary if desired.

As a result, we are interested in Machine-aided human summarization because the task of generating summary is a creative work and it cannot be totally fulfilled by the machine. That is why a better way to tackle the problem of generating the summarization is to help the human execute a piece of work more quickly. The main assignment of my Master's Thesis is to build a chain of technologies to help a researcher write a high-quality review (phrase by phrase) even if he does not understand the topic completely.

In 1980 when Martin Kay suggested the machine-aided translation approach the idea of machine-aided summarization was introduced. He [37] proposed the development of cooperative man-machine systems as a solution to the unrealistic task of fully automatic high quality translation, allowing the computer and the human translator to perform the translation tasks they are best at.

Definition 1.5. Machine Aided Human Summarization is a generation of a review in which a person receives help from a computer program during the summarization or in

other words human is aided while doing the task of summarization [45].

Definition 1.6. Human Aided Machine Summarization is a formation of a review by editing the summary generated by a computer program [45].

One of the systems which provide Human Aided Machine Summarization is Human Aided Text Summarizer "SAAR" [51]. It is built with the help of reinforcement learning and is proposed for summarizing a single document. A generated summary is shown to the user and if it satisfies the requirements it is kept as the final version of the summary otherwise a new summary is generated according to the feedback of the user in a form of keywords.

Another paper [49] is eager to help the researcher who makes summary by outlining the remarkable information from a paper and presenting it to him/her. In this case the task of linking sentences to form a coherent summary is left to the human. This approach is similar to the idea of our system. However, we work with a collection of papers instead of one and use a different methodology and tools to solve the problem.

To sum up, the motivation behind this thesis is the creation of the method to intensify the work of researchers in scientific papers analysis field.

At the moment, there are ready-made services that allow you to collect selections of scientific papers on a given topic. Some of the main services are semanticscholar.org, mendeley.com, academia.edu, dimensions.ai, aminer.org and arxiv-sanity.com. In addition, the MIPT team - arxiv-search.mipt.ru - has developed an assembly of collections of scientific papers and recommends new papers on them. Now you can assemble a selection of several dozen works on the necessary topic not in days or hours, as it was before, but literally in tens of minutes.

As you can see, there are a lot of such systems today, since we are in close contact with the MIPT team, we have a desire to build our solution in their service and help people write an essay on the topic they need in a matter of hours (rather than days) on the collection of papers. An example of how this should work is presented in Potential Impact.

1.2 The problem statement

A quantity of papers is given and the technology which offers the user a ranked list of recommended phrases has to be built. Later on these phrases are used to compose the coherent summary of the papers.

With respect to the own goals of the user our service is supposed to have various methods for summarizing papers (called prompters). In other words, it is supposed to have an ability to continue the phrase in several ways.

As a result, we are developing a system which consistently recommends phrases to the user in order to write the text of the summary.

Definition 1.7. Prompter - is the function of our system which allows the user to choose a specific summarization method.

1.3 Objectives

Therefore, we have the task of machine learning, where the input is a whole lot of scientific papers, and the output is a set of phrases arranged in a certain order using one of the

prompters (summarization method). The set of phrases is limited and various for each prompter.

This is a complex and challenging task. Hence, we divide it into the following subtasks, and also we give examples of possible prompters based on the needs of researchers.

Firstly, we need to organize papers added to the collection. Here, the training sample is how and in what sequence other papers are referred to in scientific papers. That is, with the help of solving the problem of machine learning, we are trying to find out whether the logic in this sequence is common for the entire scientific community and whether this logic can be learned. From the statement of the problem, it is obvious to use review parts of scientific publications, as authors are to build citation logic of other works. If we consider the citation order throughout the paper, it is impossible to find out some kind of dependency.

Secondly, we need to learn how to highlight the overview part in the text of the paper. In addition, we need to learn how scientific papers refer to other papers, highlight reference and inline citations and also learn how to relate them to each other.

One of the main prompters should be one that shows how this paper is referred to in other papers. Given that we know the location of inline citations, we can highlight the text fragment of the citation. Here you can use the training sample already marked out by someone, associating it with our dataset. To determine the candidates for the current fragment of the citation, we are to determine in which sentence the citation is located and in which place of the sentence. We are also to determine whether there are other citations in this fragment. The presence of a coreference between sentences implies correlation of both sentences to one citation. Finally, we need to train the classification model and make a ranking of phrases (sentences).

In addition to the previous prompter a prompter that determines which part of the paper this citation refers to can be built: Aim, Hypothesis, Method, Result, Implication, Dataset.

Given that the key ideas of the paper reflected in the abstract may become outdated and other parts of the paper become more vital for the scientific community which are not included in the abstract, it is logical to build a prompter to find the sentences in the paper that are used to write citations. In this situation, there should be a large labeled training sample by which we can:

- learn the relationship between the sentences of the paper and the way other authors refer to the original work;
- train a classification model based on various features;
- rank phrases and show it to the user.

Prompter for finding phrases which describes a paper in the best way:

- Here, first of all, we need to select the following sections of the paper: abstract, results, conclusion.
- Then we separate sentences based on coreference.
- After that we train the classification model based on the similarity of sentences with Summary or, if Summary is not presented, with Abstract. During this we use the Rouge metric (or other metrics).

- Finally, we need to rank phrases.

Also, to reflect the various key phrases of the sections it is logical to build a Generalization of tagging prompter. The definition of each category is a separate prompter in it. To build a Generalization of tagging prompter we need:

- To collect a training sample in which all the phrases of the paper will be marked with various tags: Aim, Hypothesis, Method, Result, Implication, Dataset.
- Train the model of extractive summarization.
- Rank phrases and show it to the user.

Definition 1.8. Coreference - is a text identifier indicating that near sentences relate to one idea/ topic [36].

Definition 1.9. Citation based summarization - is a procedure of text generation which is formed by utilizing a set of citations to a referenced paper [52, 53].

1.3.1 The description of models which a similar to our prompters

Definition 1.10. Citing Paper(CP) is a paper that cites another paper [32].

Definition 1.11. Reference Paper(RP) is a paper that is cited by another paper [32].

Definition 1.12. Citation text is the sentence(s) in a citing paper that contains the citation and conveys the authors' discussion of the citation. The citation text may consist only of the sentence containing the citation, or may include one or more sentences before and/ or after the citation. A citation text may also consist of a portion of the sentence containing the citation [32].

Since we use citations to do the summarization let us compare our prompters with already existing methods.

Early work in citation based summarization [1, 21, 46, 52] aimed to extract the most relevant citing reference paper sentences.

Definition 1.13. Citation sentence (citing sentence) is a sentence that cites the reference paper. Also, a citation sentence can be viewed as a short summary of the reference paper written from the citing authors' perspective [66].

Hence, a collection of citation sentences reflects the impact of the reference paper on the research community [21]. While citation sentences provide the community's views of the reference paper, prior works [42,59] point out issues in using citation sentences directly for summarization. In citing sentences, the discussion of the reference paper is often mixed with the content of the citing paper or with the discussion of other papers cited jointly, containing much irrelevant information. To address such issues, recent work [17–19,33,42] considers cited text spans-based summarization, where they identify a set of text spans (a set of sentences) in the reference paper that its citing sentences refer to, and perform summarization on the identified text spans. This way, while the summary consists of words in the reference paper, it reflects the research community's insights. In this work [66] advanced multi-document summarization system is presented. A Graph Convolutional

Network (GCN) on the relation graphs with sentence embeddings obtained from Recurrent Neural Networks as input node features are employed there.

One of the extractive summarization sections is Automatic Survey Generation. We can consider the related work as a survey for a specific field.

The related work section can be mainly regarded as a survey. A significant part of work which is related to Automatic Survey Generation exploits extractive summarization approach. The earliest ReWoS method used a hierarchical set of keywords that describes the topics of the target paper to extract related works [27]. The later work [28] used a global optimization structure (with help of PLSA model) to create related work.

Recently, [12] created related works from a graph-based comparative summarization method which worked with papers (locate at related works) from references . After that, they used the minimal Steiner tree to control the generation, extracting the least amount of sentences to cover the discriminated nodes. This approach does not take into account the content of target and reference papers, leading to topic deviation.

[64] developed a data-driven neural summarization model which combined context-driven attention mechanism to create an appropriate working section. They created a directed graph containing heterogeneous relations among kinds of objects, such as papers, authors, keywords and places, and developed an attention mechanism that focuses on the contextual relevance in the target paper being written and the graph. For each candidate sentence, a label of 0 or 1 was assigned after optimization of the target logarithmic probability.

Toc-RWG [63] with QueryTopicSum (a LDA-style model to characterize the generative process of both the scientific paper and its reference paper) and the identified CTS (Cited Text Spans) as candidate sentences, an optimization framework based on minimizing KL divergence is exerted to select the most representative sentences for related work generation.

HSDS [35] explores an abstractive method for automatic survey generation with the help of Seq2seq model based on Dual Supervision.

Another work [2] tries to reduce the redundancy of citation sentences from multiple related work sections and enhance the readability of the generated summary by investigating a semantic graph-based approach and cross-document structure theory.

Surveyor [34] summarization algorithm combines a content and discourse of source papers when generating survey papers. CitationAS [62] uses topic modeling to highlight the topic of the citation text and by calculating various features of the sentence, ranks them and compiles summary. And finally, IBM Science Summarizer [22] builds extractive unsupervised query-focused multi-document summarization approach which provides users with summaries for every subsection of paper. Our difference from them is that they focus on keeping updated on current work and we are committed to prepare the summary for a research project / grant request.

1.4 Potential Impacts

The whole work represented in the pipeline was made from the beginning to the end. Starting from the ranking document collections and ending with provision a summary to a user. Our citation based summarization from reference papers method out-performed current methods on The CL-SciSumm Shared Task 2018.

Our main contribution is the definition of a scheme that allows us to add various

summarization methods to the general summarization service.

The potential impacts of this model lie on encapsulating summarization tools to https://arxiv-search.mipt.ru/ that is an exploratory search and recommendation system for the researchers that regularly seek for scientific papers in arXiv.org. The scheme of how our summarization system should work on this site is as follows:

1. It is assumed that a collection of necessary scientific papers has already been assembled by the user.

2. This collection is ranked in accordance with the order in which papers are submitted for summarization (preparation of the final report).

3. Summarization is performed for each paper depending on which method of summarization the user has selected.

4. The user edits the resulting summary according to his/her needs and adds it to the summary overview. Also a user have an opportunity to click on the paper from ranked collection. After that a new window in the browser with the full text of the paper is opened.

5. After going through all the papers in the collection the user saves the received report.

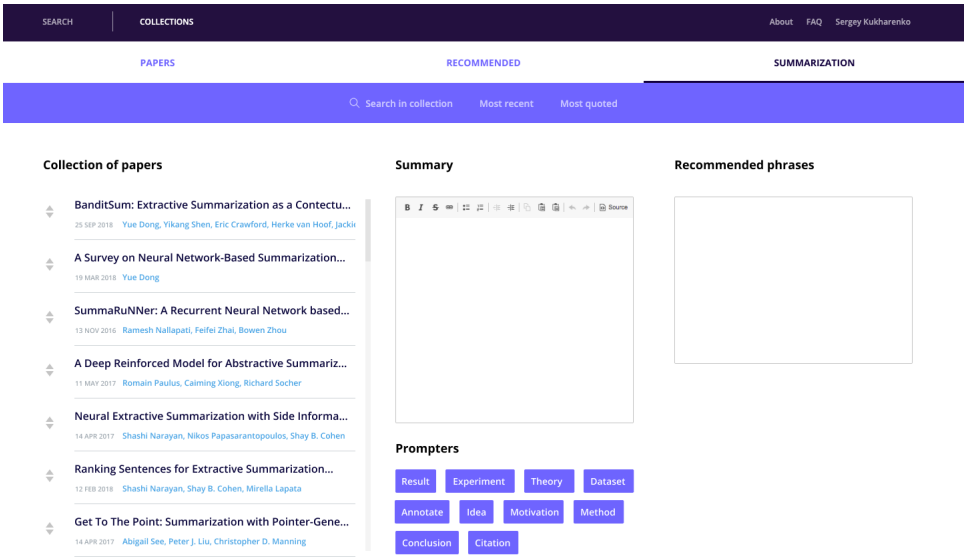

Рис. 1: The outcome of the first and the second items according to our scheme of summarization

1.5 Thesis Structure

The following chapters of thesis work is organized as follows:

Chapter 2 presents a methodology developed to solve this problem, which is divided into subtasks. For each subtask, there is a description how the problem is defined and the solution algorithm as well. The sequence of the description of the subtasks makes it possible for the reader follow our logic of reason to solve the whole problem. What has

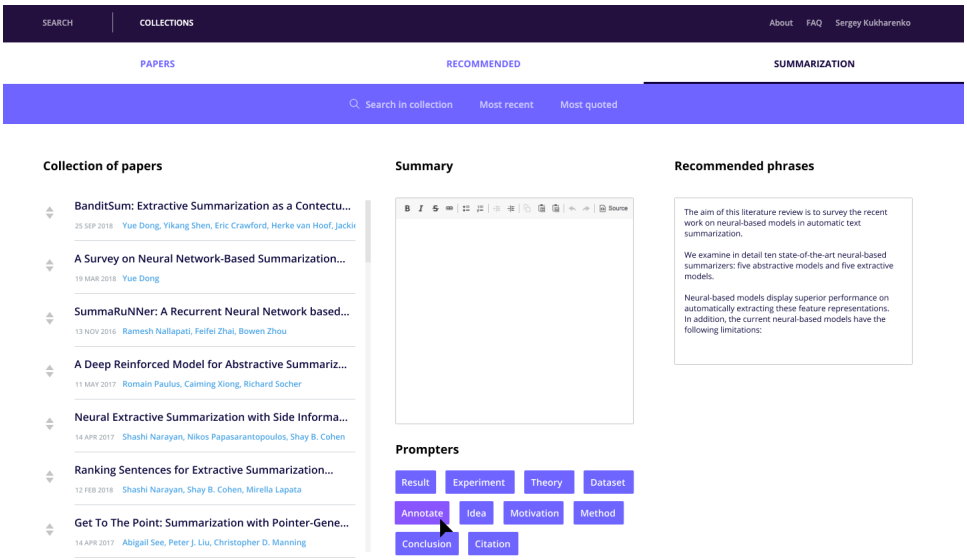

Рис. 2: The outcome of the third item according to our scheme of summarization

been done: the citation-based summarization method based on the extraction of the most relevant citing reference paper sentences and an approach to maintain the best order of papers in the report have been developed.

In *Chapter 3* we present the results of our experiments. And we consider the two main approaches to generate the best summary. In this chapter you can see the results of the work where we compare our models with the best model of 2018. It is worth noting that some of our models were better than the model of 2018 according to the Rouge metric.

In the result, Chapter 4 summarizes the research work and focuses on the main conclusions gained. It also presents an outlook for upcoming challenges of this thesis research work.

2 Methodology

This chapter covers a methodology shown in the pipeline of the master's thesis.

2.1 Pipeline

In order to build our machine-aided human summarization system we are to form a project pipeline. To describe it, we use the decomposition approach and divide a complex technical system into blocks with their subsequent description. Each block is an algorithm for solving a specific problem which receives the necessary dataset at the input and at the output we get a result. We make an assessment of the quality of the algorithm of each block using a specific criterion.

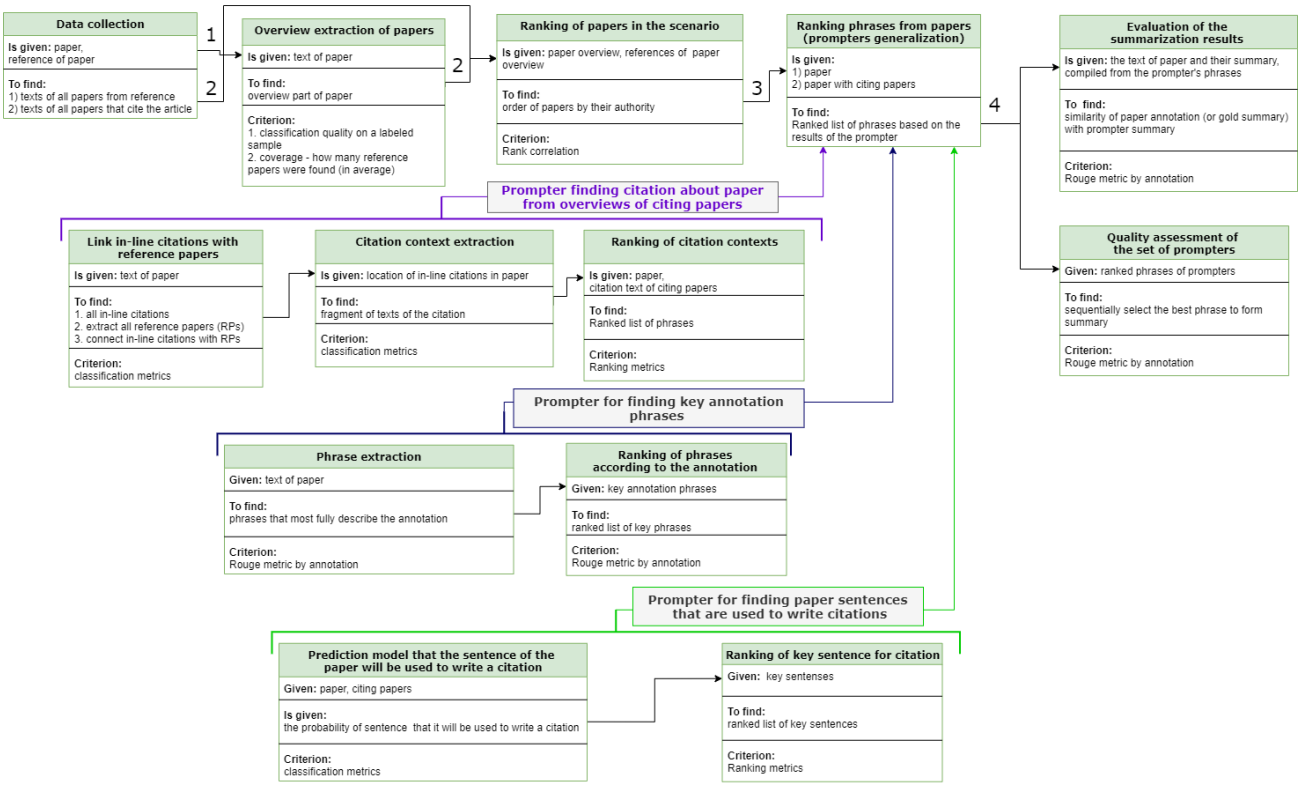

Рис. 3: Work pipeline of Master Thesis

The block **Ranking phrases from papers** is a generalization of prompters, in other words the prompters below are the elements of the unit. The arrows marked the number are to be followed along the main pipeline.

One of the main advantages of this approach lies on its novelty and the ability to add new summarization methods in the pipeline.

We divide the task into modules. For the execution of each block we use different datasets. Also there is one dataset that we use to accomplish two different modules. Every dataset is described before being used.

However, we wish to create one common dataset for the entire pipeline, it can be built as follows:

- We have paper texts.

- Then we highlight Reference papers and overview part of the papers.
- Finally, we take only those references that are in the overview part of the paper.

Its final form is as follows:

$$D = \{(x_i, y_i)_{i=1}^n\}$$

where x – unordered set of reference papers, y – overview sections of scientific papers with tagged phrases. The assembly of this dataset will be done in the future.

2.2 Data collection for extracting of the overview part and ranking papers from the collection

Despite the fact that for our task there is no ready-made dataset we examine various datasets with a large number of scientific papers that can be used as a research part of our task:

Corpus	Year	#P.	Cit.cont.	Ref.	Lin.	Domains
S2ORC(PDF-parse) [40]	2020	8.1M	full text	yes	S2ORC(full)	multi
S2ORC(LaTeX-parse) [40]	2020	1.5M	full text	yes	S2ORC(full)	phys,math,CS
PubMed Central (OAS)	2020	2.6M	full text	yes	PubMed	bio,med
ACL-AAN [54]	2013	25k	full text	no	ACL Anthology	CS,CL
unarXive [56]	2020	1M	snippets	no	MAG [58]	phys,math,CS
RefSeer [29]	2015	1 M	snippets	no	CiteSeerX [10]	multi

Таблица 1: Overview of existing datasets [40]
P.= Papers with body text; Cit.cont. = Citation context; Ref. = References to tables / figures / equations; Lin. = Linked to graph;

A contextual citation graph like ACL Anthology Network (AAN) covers papers in the field of computational linguistics. It is built on the basis of the ACL Anthology [6] and includes 24.6 thousand papers, manual citation information is added to it as well.

One of the largest corpuses is the central PubMed Open Access Corps including 2.6 million papers in the biomedical field with citations related to PubMed Identifiers.

The RefSeer is the most frequently used evaluation dataset for citation-based tasks without integrating metadata provided by sources outside the PDF. Its citation context contains 400 chars.

Later on unarXive is presented as a contextual citation dataset which is built with using 1.0M arXiv publications. LaTeX source is used to extract a text, citation spans and bibliography entries which are linked to papers in the Microsoft Academic Graph [58]. The citation context is provided through extracted snippets but without Reference parses.

S2ORC is a large contextual citation graph dataset which contains 81.1M papers, 380.5M citation edges, and associated paper metadata. The full text and inline citations related to the references are available for 8.1M papers. 1.5M papers of them also contain full-text LaTeX parses, from which authors extract the source text of tables and mathematical formulas.

The following features that are implemented in the S2ORC dataset:

- There is a full text of papers for a large number of them. This texts are compiled from a PDF file and / or latex source files which are used for publication on arxiv.org.

Total papers	81.1M
Papers w. PDF	28.9M (35.6%)
Papers w. bibliographies	27.6M (34.1%)
Papers w. GROBID full text	8.1M (10.0%)
Papers w. LaTeX full text	1.5M (1.8%)
Papers w. publisher abstract	73.4M (90.4%)
Papers w. DOIs	52.2M (64.3%)
Papers w. PubMed IDs	21.5M (26.5%)
Papers w. PMC IDs	4.7M (5.8%)
Papers w. ArXiv IDs	1.7M (2.0%)
Papers w. ACL IDs	42k (0.1%)

Таблица 2: Statistics on paper provenance in S2ORC [40]

- Inline cite spans are highlighted in the text of paper.
- Bibliography entries are highlighted and an ability to link papers and a create citation graph is presented.
- The majority of inline citations are connected with their reference papers.
- The text is divided into paragraphs and some of them have the title.
- Each paper inflates metadata and an abstract.
- The dataset contains papers from different domains.

As the main dataset for conducting the research we use S2ORC.

The S2ORC dataset weights approximately 1 TeraByte that is why I take only Anthology of Computer Linguistic papers from it. After the cleansing: removing the duplication of publications, deleting publications that have no text - we create a citation graph with 40k paper nodes.

Also we use Semantic Scholar Open Research Corpus [3], the citation graph of https://www.semanticscholar.org/ papers, to check how proficient the citation graph is compiled for our dataset and for the creation of the author's citation criterion.

2.3 Overview extraction of papers

As it has been mentioned, we are eager to learn how to detect the overview part of the text in the paper because this helps us to create a training sample for the Ranking of papers in the collection block. We are also willing to detect sections of papers where the author, along with citing, briefly describes previous scientific papers on this topic. In the future, these texts will be used in the work of prompters.

2.3.1 Problem statement

The input is the text of a paper. Our goal is to find the overview part of the paper, which briefly describes other papers. To check the quality of the classification on the marked sample is our criterion.

The chapter , where the author along with citing briefly describes previous scientific works on this topic, is called Related Work. Also, some authors may describe previous works in the Introduction or Methods chapters. However, not all the papers have introduction and related work, which can be considered as an overview because they may contain some information that is not relevant to the description of the previous works.

Therefore, we cannot simply divide the paper into various sections and, based on the section name, relate it to the description of the previous works. For this reason, we want to set the task of machine learning which highlights text parts (sections / paragraphs) with a description of the previous works based on the signs of a part of the text (sections / paragraphs).

The sections in our dataset are paragraphs of the text. Taking into account that in order to determine whether a paragraph is the overview or not, we are to draw a sample of our own, therefore we consider the paragraph to be the overview if it is in the chapter that is called Related Work.

2.3.2 Models description

As training and test sets we use a subset of ACL papers that have section titles. The target texts are those whose title contains one of the following words or phrases: Related Work, Background, Previous Work.

In the Baseline model we assign that the section is the overview if it has the maximum number of citations among other sections of the paper. We also assign a section as the overview if it contains half of this maximum number of citations.

For machine learning models we form the following features:

- citation density = $\frac{\text{number of citations}}{\text{paragraph length}}$
- The number of consecutive sentences which include at least one citation
- The normalized position of the chapter in the paper = $\frac{\text{number of position} + 1}{\text{count of positions}}$
- An average position of inline citations = $\frac{\frac{1}{n}\sum_{i=1}^{n} \text{start position of inline citation}}{\text{length of section}}$, where n - count of inline citations,

The Gradient Boosting model shows itself to be the best one [13].

2.4 Ranking papers in the collection

Based on the fact that the user of our system does not have to understand/be an expert in the field of science from which summary is written the system should advise the user in what order to mention papers in the report.

As mentioned, as a training sample we use how the citations are located in the overview part of papers since authors are to build a citation logic for other works. If we consider the citation order throughout the paper it is impossible to find out any dependence.

2.4.1 Problem statement

We have an overview of paper and we know in what order the references in the text are located. Our goal is to find order of papers by their authority. The quality of ranking is

measured by the number of defective pairs (rank correlation of kendell) comparing with the order of mentioning papers in the review.

We do not use other measures of ranking quality since these are metrics for the search engine and try to take into account that the user rarely looks at the second, third and other search results pages, and we have to take into account the order relation.

Definition 2.1. Kendall rank correlation coefficient. Let (x_i, y_i) be a set of observations of the joint random variables X and Y respectively,such that all the values of (x_i) and (y_i) are unique. Pairs (x_i, y_i) and (x_j, y_j) where $i < j$:

- Concordant if both $x_i > x_j$ and $y_i > y_j$;or if both $x_i < x_j$ and $y_i < y_j$
- Discordant if both $x_i > x_j$ and $y_i < y_j$;or if both $x_i < x_j$ and $y_i > y_j$
- If $x_i = x_j$ and $y_i = y_j$, the pair is neither concordant nor discordant

$$\tau = \frac{\text{(number of concordant pairs) - (number of discordant pairs)}}{C_n^2}$$

2.4.2 Models description

In the Baseline model, we rank papers by year in ascending order.

For the ranking model we built the following features:

- The year of paper publication.
- The number of paper citations.
  It is the number of paper citations among all S2ORC papers which have a full text and references, this number is equal to 5M.
- The citation of a journal or conference.
  It is a number of journal or conference citations among all S20RC papers which have full texts and references, this number is equal to 5M.
- The author citation.
  It is the number of citations to the work of this author in other papers. To create this feature we combine our dataset (S2ORC) and Semantic Scholar Open Research Corpus dataset [3] by paper id. Then we uploade influential citation count using Semantic Scholar API.
- Presence of identifier (ACL, PubMed, DOI, arXiv).

Let's introduce a notation:

- Let $Q = \{q_1, \ldots, q_n\}$ be the set of groups (an overview parts of ACL papers)
- $D_q = \{d_{q1}, \ldots, d_{qm}\}$ – set of objects (reference papers) retrieved for a group (ACL paper) q
- $L_q = \{l_{q1}, \ldots, l_{qm}\}$ – relevance labels for the objects from the set D_q (a normalized order of our papers)

Every object d_{qi} is represented in the vector space of features, describing the associations between the group and the object.

So, every group is associated with the set of objects. For example, the group is a query (the overview of ACL paper) and object is a paper (reference papers of ACL paper) if we rank documents for a search query.

The goal is to learn the ranking function $a = a(d_{qi})$, such as the ranking of objects d_{qi} for all groups from Q based on their scores $x_{qi} = a(d_{qi})$, as close as possible to the ideal ranking from the editorial judgements l_{qi} (That means that we do the ranking to order papers in our collection in the best way).

We know the order of reference papers in the overview of a paper. Next, we compare papers from references in pairs: $i \prec j$ - i less relevant than j (i locates before j in the text of paper). To learn the paper order in references we penalize our model if the ranking function establishes that j is less relevant than i.

So, we have to minimize empirical risk:

$$Q(a) = \sum_{i \prec j} [a(x_{qj}) - a(x_{qi}) < 0] \leq \sum_{i \prec j} \mathcal{L}(a(x_{qj}) - a(x_{qi})) \to min$$

where $a(x)$ - ranking algorithm; $Margin(i, j) = a(x_{qj}) - a(x_{qi})$;
$\mathcal{L}(M)$ - decreasing continuous function of Margin $Margin(i, j)$:

- $\mathcal{L}(M) = (1 - M)_+$ - RankSVM
- $\mathcal{L}(M) = exp(-M)$ - RankBoost
- $\mathcal{L}(M) = \log(1 + e^{-M})$ - RankNet (like Logistic Regression)

As it can be seen Loss function can be chosen in different ways and according to them distinct ranking methods are to be implemented.

As a ranking model we train CatBoost using pairwise approach [20]. The algorithm of the model is as follows:

1. To subsample the overview of ACL papers (acl-papers) that contain at least 90% of reference papers in S2ORC dataset.
2. To upload all reference papers from acl-papers (out-papers)
3. To calculate all the features for out-papers and concatenate them to acl-papers, i.e. to form a feature space.
4. To define the measure of relevance as the order of reference papers (RPs) from the overview. Then to normalize this order as our acl-papers have different number of RPs.
5. To train Gradient Boosting model using pair-wise approach which minimizes the negative loglikelihood:

$$- \sum_{i,j \in Pairs} \log \left(\frac{1}{1 + e^{-(a(x_{qi}) - a(x_{qj}))}} \right)$$

In the future the topic similarity between papers is to be taken into account to locate similar papers nearby.

$$topic\ similarity = KL(ref.paper || collection)$$

2.5 Prompter for finding paper sentences that are used to write citations

After ranking our papers in the collection we can move to the summary generation for each paper using various summarization methods (prompters).

Providing that the key ideas of the paper reproduced in the abstract may become out-of-date and other parts of the paper become more crucial for the scientific community which are not included in the abstract, it is reasonable to create a prompter to pick the sentences in the paper for writing citations.

2.5.1 Problem statement

We use papers with papers citing them and annotations which show relations between citing sentences and sentences from the original papers (i.e. annotations show which sentences have a greater impact on citations).

So, we are to detect paper sentences that are used to write citations. Quality comparison of generated machine summary with gold (made by human) summary using Rouge metrics is our criterion.

2.5.2 Dataset

To solve this problem I use the CL-SciSumm 2018 corpus dataset which contains:

- 40 annotated sets of citing and reference papers from a computer linguistic domain.
- Each Reference Paper includes ten or more Citing Papers that contain citations to the Reference Paper.
- Also every Reference Paper contains Annotation file which connects citing sentences and reference spans and also show their positions in papers.
- Additionally, the dataset provides three types of summaries for each Reference Paper:
  - an abstract paper, written by the authors of the research paper.
  - a community summary, collected from the reference spans of its citing sentences.
  - a human-written summary, written by the annotators of the dataset.

2.5.3 Model description

Firstly, we make connected data structure of Reference and Citing papers. Also we read the Annotation file for connecting sets of citing sentences and reference spans.

Secondly, it is possible to compute a set of features for each reference paper, for each citing sentence and each sentence of the reference paper. In other words, between each sentence of RP and each sentence of CP we calculate the following groups of features:

- Cosine similarity measure between embeddings (model is trained on RP sentences and its CPs sentences) of sentences:

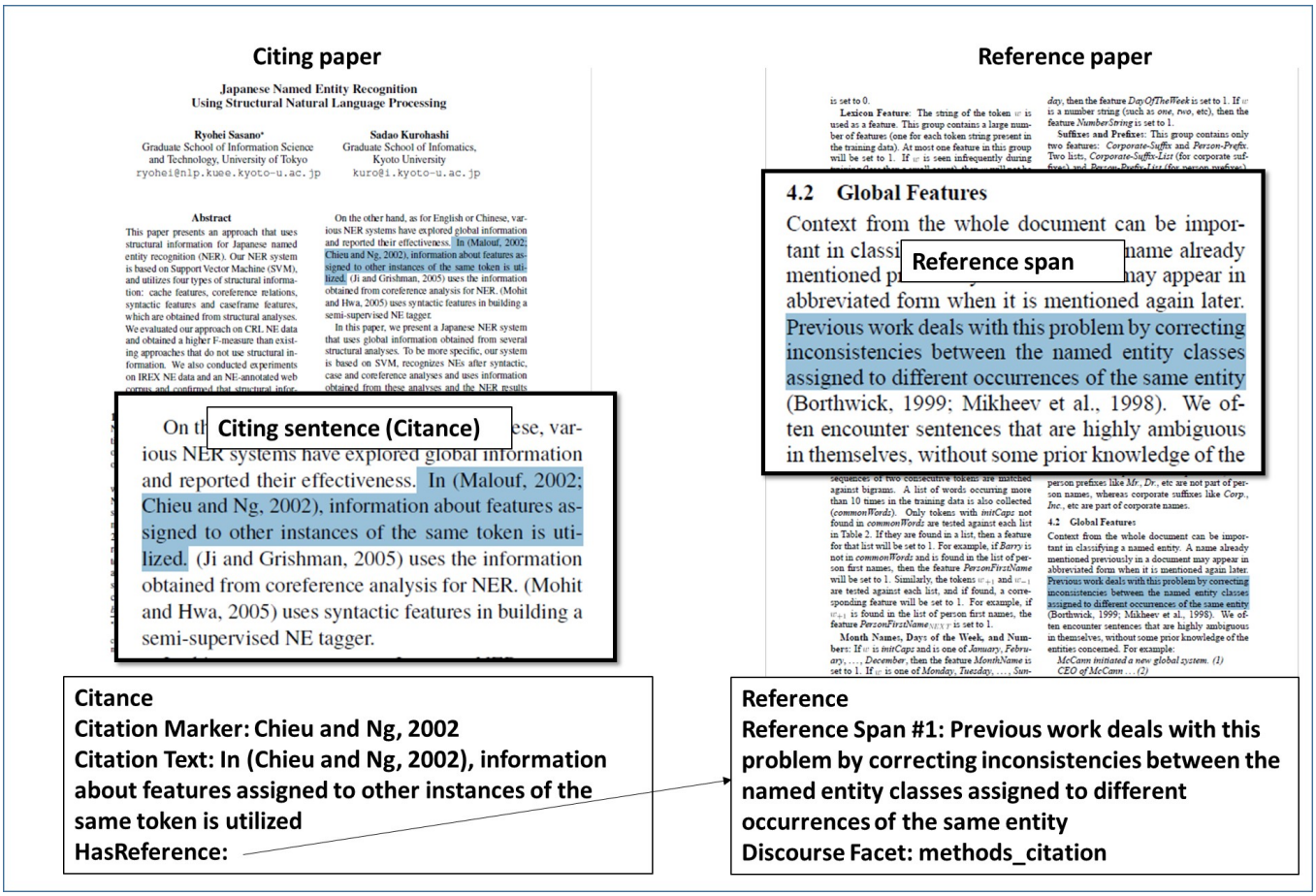

Рис. 4: The CL-SciSumm 2018: The dependancy between reference paper and citing paper via annotation file [33]

  - TF-IDF model cosine similarity
  - Latent Semantic Indexing model cosine similarity [26]
  - Latent Dirichlet Allocation model cosine similarity [7]
  - Hierarchical Dirichlet Process model cosine similarity [61]
- Cosine similarity measure between averaged (by words) pretrained embeddings of sentences.
  - W2V model cosine similarity [44]
  - Word Mover's Distance between embedded word vectors [38]
- Comparison using N-grams:
  - Number of common bigrams
  - Rouge scores (-1,-2,-l f1) [39]
- Sequence Matcher ratio [55]

$$\mathcal{D}_{ro} = \frac{2\mathcal{K}_m}{|S\_RP| + |S\_CP|}$$

where S_RP is a sentence of a reference paper, S_CP is sentence of a citing paper, $\mathcal{K}_m$ (the number of matching characters) is the longest common substring (LCS) plus recursively the number of matching characters in the non-matching regions on both sides of the LCS

- Positional features:
  - The position of the sentence in the RP = $\frac{\text{number of sentence position}}{\text{count of sentences in the RP}}$
  - The position of the sentence in the section of the RP
  - The position of the section in the RP

Thirdly,these features are used to train a classifier to predict if a sentence of a reference paper is a reference span or not.

Finally, given all the probabilities for a sentence of the reference paper of being a reference span or not, a global score for a reference span candidate of the RP is computed by summing all its probabilities from different citing papers.

For generating summary the model chooses sentences with the highest score until the limit of words is reached (for the CL-SciSumm summarization task the length of the summary exceeds 250 words).

2.6 Prompter finding citations about papers from overviews of citing papers

One of the main prompters is the one that shows how this paper is referred to in other papers.

2.6.1 Problem statement

The text of a paper and the text of a citing papers is given to us.

And we have to find ranked list of citing sentences that reproduce the description of the work by other authors in the best way.

As a criterion we take a quality comparison of a generated machine summary with the gold (human) summary by Rouge metric. The block of the summary evaluation is described below.

2.6.2 Model description

Since we know the location of inline citations, we can highlight the text fragments of the citation. Here you can exploit the training sample already marked out by somebody connecting it to our dataset.

To select the candidates for the current fragment of the citation, we are to work out in which sentence the citation is located and in which part of the sentence. We are also to define whether other citations in this fragment exist. The presence of a coreference between sentences means a connection of both sentences and a citation. Ultimately, we have to train the classification model and make a phrase ranking.

On top of this prompter we create a prompter that defines which part of the paper this citation refers to: Aim, Hypothesis, Method, Result, Implication, Dataset. As a training sample SciCite dataset of citation intents [16] can be used.

2.7 Prompter for finding key annotation phrases

2.7.1 Problem statement

A text of paper is given to us. And we have to rank key sentences of the paper that describe it in the best way. The comparison of a generated machine summary with the gold (human) summary by Rouge metric is our criterion. The summary assessment block is described below.

2.7.2 Model description

The prompter for finding phrases which describes a paper in the best way:

- Here, first of all, we need to select the following sections of the paper: abstract, results, conclusion.
- Then we separate sentences based on coreference.
- After that we train the classification model based on the similarity of sentences with Summary or, if Summary is not presented, with Abstract. During this we use the Rouge metric (or other metrics).
- Finally, we need to do a phrase ranking from probability of sentences.

2.8 Evaluation of the summarization results and prompters

First of all, let us consider how we evaluate a generated summary (sentences) by comparing it to the gold (human) summary. Further we determine the quality of work of a set of prompters and find the most optimal summary of several ones(generated by different prompters).

2.8.1 Summary evaluation

Problem statement
We have the text of the paper and its summaries collected from the sentences generated by prompters. And we are to compare annotations of papers (or summaries) with the gold (human) summary using ROUGE metric.

Metric description

The vast majority of summarization works use the Rouge metric as an automatic comparison of a generated machine summary with the gold (human) summary [39].

$$ROUGE_N = \frac{\text{number of overlapping N-grams}(human_{summary}, system_{summary})}{\text{number of N-grams in } human_{summary}}$$

$$ROUGE_L = \frac{(1+\beta^2)R_{LCS}P_{LCS}}{R_{LCS}+\beta^2 P_{LCS}}, \quad R_{LCS} = \frac{LCS(X,Y)}{|X|}, \quad P_{LCS} = \frac{LCS(X,Y)}{|Y|}$$

where $LCS(X,Y)$ the length of a longest common subsequence of summaries X and Y, $\beta = \frac{P_{LCS}}{R_{LCS}}$.

However, it should be noted that automatic summarization metrics have serious limitations:

- They only assess a content selection and do not account for other quality aspects, such as fluency, grammar, coherence, etc.
- To assess content selection, they rely mostly on lexical overlap, although an abstractive summary could express the same content as a reference without any lexical overlap.
- Given the subjectiveness of summarization and the correspondingly low agreement between annotators, the metrics are designed to be used with multiple reference summaries per input.

Therefore, as an additional checking of the summarization quality it is better to further fulfill manual comparisons of optional summaries, but it is time-consuming and still a common standard does not exist. That is why it is to be done in the future.

2.8.2 Quality assessment of a set of prompters

The summary researcher can generate as follows:

1. From the summaries generated by different prompters we choose the most relevant one.
2. We collect summary sequentially sentence by sentence, where (at each step) the best sentence is selected from the top-k ranked sentences generated by prompters.

Moreover, the second approach gives an opportunity to estimate the quality of prompters which is determined by the number of sentences generated by prompters that are included in the summary.

Problem statement
Ranked sentences generated by prompters are given. We are eager to sequentially choose the best sentences to form the final version of the summary. As a criterion we use ROUGE metric by selecting the best sentences (from top-k generated sentences by different prompters) according to it. So, we are willing to evaluate the quality of prompters' work rather than the quality of summaries or human behavior while choosing a summary version. Hence, we train our system to offer the best summary.

Model description

The algorithm is as follows:

1. Follow the ranked sequence of papers in the collection.
2. Set the number of sentences k that we need to extract.
3. Generate and rank top_k sentences using various prompters for each paper.

4. Compare all the top_k sentences with the gold (human) summary and choose those that have the best Rouge metric.

5. Remove the selected sentence from the sample and increase by one the number of prompter exploitations from which the most suitable sentience was selected.

6. Repeat steps 4 and 5 until we form a summary of top_k sentences.

7. Get the final version of the summary and a list with the number of prompter exploitations.

3 Experimental results

In this chapter numerical results obtained for each fulfilled module are described.

3.1 Dataset

3.1.1 Calculation of covering for the main dataset

It is necessary to mark out a subsample of the dataset S2ORC where the lists of citing papers and reference papers are as less thinned as possible (i.e. they are presented in the dataset with full text).

The following features for all papers of the two subsections (ArXiv, ACL) are calculated:

- #bibs - the amount of recognized Reference papers;
- #bibs-with-links - the amount of recognized Reference papers contained in dataset S2ORC;
- bibs-with-links-in-dict - the amount of recognized Reference papers contained in data structure **in** which includes papers and papers referring to them;
- %-in-dict2bibs - the proportion of the number of Reference papers contained in data structure **in** to the number of recognized Reference papers;
- %-in-dict2bibs-w-l - the ratio of the number of Reference papers contained in data structure **in** to the number of Reference papers contained in the dataset S2ORC;

The following results are:

subsample	aver. %-in-dict2bibs	aver. %-in-dict2bibs-w-l
ArXiv	38.2%	44.8%
ACL	42.4%	57%

Таблица 3: Calculation of covering for the full text of papers

subsample	aver. %-in-dict2bibs	aver. %-in-dict2bibs-w-l
ArXiv	40.6%	48.3%
ACL	44.5%	59.6%

Таблица 4: Calculation of covering for an overview part of papers

This result is considered admissible. And in the future every missing papers are to be downloaded in a PDF format by using API SemanticScholar and converted by using GROBID parser into the text mode.

3.1.2 Overview extraction

Unfortunately, only 3500 out of 38 000 papers provided in the dataset contain section (paragraph) titles. Hence, we use a subsample of this papers instead of the whole dataset. The subsample was splitted into train and test in proportion of 8 to 2. The titles linked to the related sections were reassigned by the following method:

- A section header is assigned as Related Work, if a title includes any of the following names: Related Work, Background, Previous Work, Overview;
- Dataset: Data, Dataset, Corpus, Corpora, Corpuses
- Setup: Setting, Setup

As for Baseline model, a concrete section is called as the overview if it covers the largest number of references amongst other paper sections.

We also trained Gradient Boosting [13] model on generated features using 5-folds cross-validation.

method	accuracy
baseline	61%
Gradient Boosting	82%

Таблица 5: Classification quality of an overview part extraction

3.1.3 Ranking papers in the collection

A subsample of the overview parts of acl-papers, containing in the whole dataset not less than 90% of Reference Papers, includes 7250 papers. The number of reference papers from acl-papers (out-papers) is 27 000 papers.

As a means of metric Kendall rank correlation coefficient(τ) is employed as it sets the order relation between reference papers. The values of this coefficient lie between -1 and 1 and indicate strong disagreement and strong agreement respectively.

The table shows the results of the following models:

- Baseline model ranking reference papers of acl-paper in ascending order according to the year.
- CatBoost model with pair comparisons that minimizes the negative loglikelihood (PairLogit):

$$-\sum_{i,j\in Pairs} \log\left(\frac{1}{1+e^{-(a(x_i)-a(x_j))}}\right)$$

metric	baseline	CatBoost(PairLogit)
τ	0.1	0.48%

Таблица 6: Quality of the ranking of papers in the collection

3.2 Prompter for finding paper sentences that are used to write citations

As CL-SciSumm 2018 dataset provides three types of summaries for each Reference Paper:

- An abstract paper written by the authors of the research paper.
- A community summary collected from the reference spans of its citing sentences.
- A human-written summary written by the annotators of the dataset.

Diverse models are constructed according to the amount of taken features and different classifiers. The models shown in figures below are horizontal curve. For comparison purposes models based on different types of summary and recognised best in 2018 are taken into account.

Using classifiers:

- SVM with rbf;
- Gradient Boosting (Xgboost, CatBoost);
- Multi-layer Perceptron.

In the case of total system summary the model chooses sentences with the highest score until the limit of words is reached.

In the case of sytem+human summary the model creates 3 such summaries (from the ranking sentences) and the best one is chosen by Rouge metric. In the future the selection work will be done by a user.

We let our system create sytem+human summary as the user might not like the generated summary or the user might be eager to look through another summary in order to choose the best one. As a matter of fact, the process of choosing the best one from the three summaries with the help of Rouge metric is seen as modelling the user's behavior to determine the best summary (currently it is fulfilled by a software and in the future it will be done by a user on https://arxiv-search.mipt.ru/).

As a result, we can model optimal behavior of a human who is willing to choose a summary from several generated ones by prompters and streamline the quality of their work.

We prefer Rouge-2-f1 score as it is commonly used in the CL-SciSumm Summarization Task [33] of the competition.

The results for total system summaries are the next:

- The best features which make an improvement for all models are w2v, wmd, lda and seq-match.
- The best classifiers on different evaluation summaries are multilayer perceptron or catboost.
- Our summarization model (Multi-layer Perceptron with basic features+hdp+lda) works better on community summaries because its collected from the reference spans of its citances.

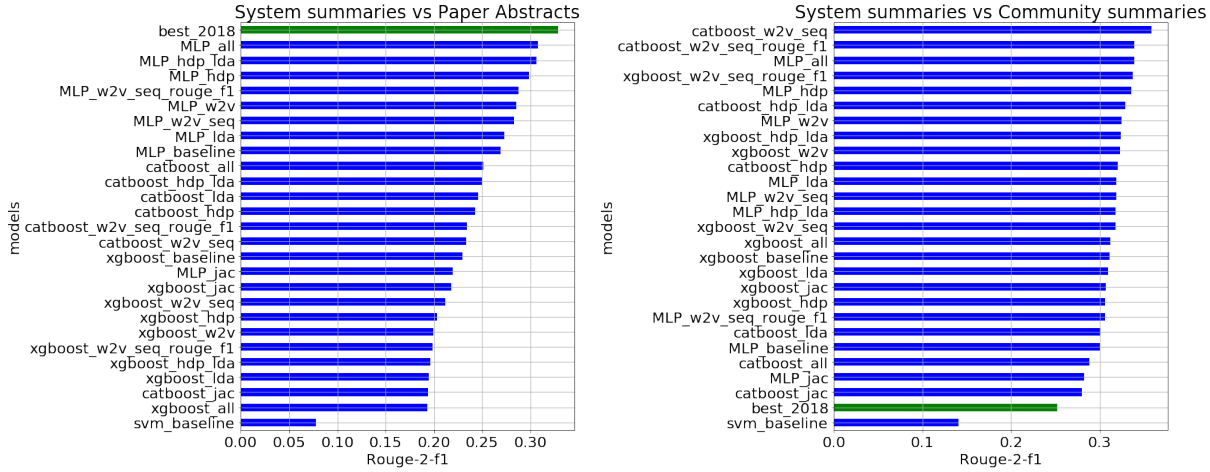

Рис. 5: Result of the comparison of the generated total system summary, paper abstract and community summary

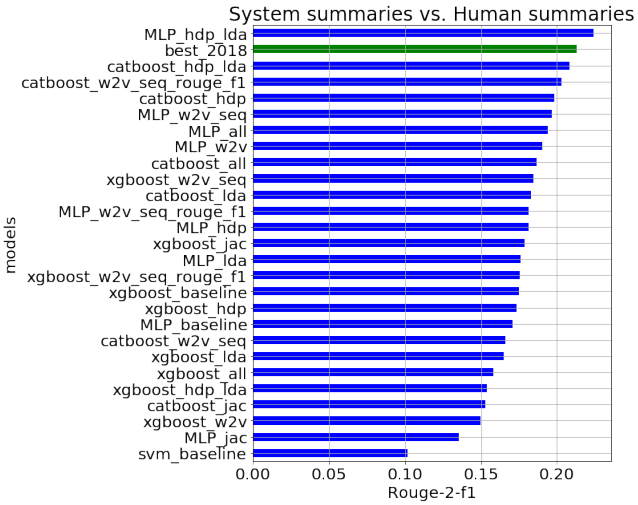

Рис. 6: Result of the comparison of the generated total system summary and human summary

- Our summarization model works better than the best_2018 model on community and human summaries because features of the best_2018 model are focused more on abstract section of the Reference Paper.

Results for system+human summaries are the next:

- Rouge metric for system+human summaries are 15-19% better than for total system summaries.
- the best total system classifier (Multi-layer Perceptron) is the same as the best system+human classifier.

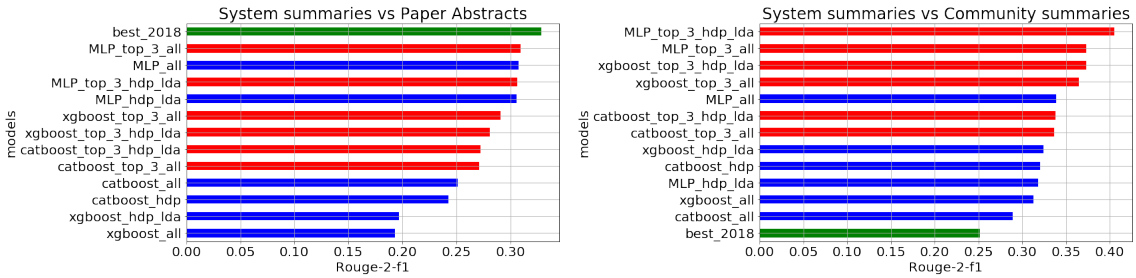

Рис. 7: Result of the comparison of the generated system+human summary, paper abstract and community summary

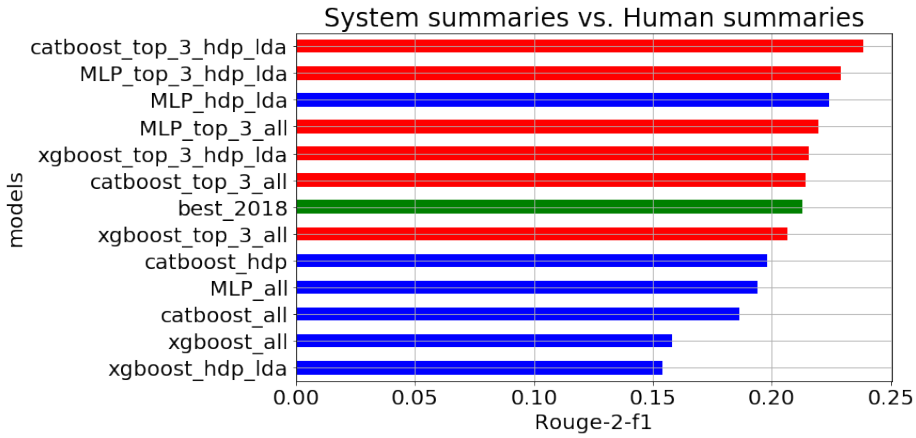

Рис. 8: Result of the comparison of the generated system+human summary with human summaries

4 Conclusions and Future Challenges

4.1 Conclusions

The following results were achieved:

- We developed the novelty methodology for generating the overview summary via multiple summarization methods.
- The whole work was made from the beginning to the end: starting from ranking papers in the collection to citation based summarization.
- The citation based summarization from reference papers (for finding paper sentences that are used to write citations) achieved excellent results.
- The new approach of quality evaluation of different summarization methods was given.

4.2 Future Directions and Challenges

Despite the time limitations, we managed to go through all the work presented in the pipeline from the beginning to the end except for the experimental validation of several prompters(multi-document summarization methods). We consider the Future work as follows:

- to improve achieved results;
- to add new prompters and implement prompters that were described without experimental results;
- implement our solution on https://arxiv-search.mipt.ru/.

Список литературы

[1] Amjad Abu-Jbara and Dragomir R. Radev. Coherent citation-based summarization of scientific papers. In *ACL*, 2011.

[2] Nouf Ibrahim Altmami and Mohamed El Bachir Menai. Semantic graph based automatic summarization of multiple related work sections of scientific articles. In *AIMSA*, 2018.

[3] Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew Peters, Joanna Power, Sam Skjonsberg, Lucy Wang, Chris Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. Construction of the literature graph in semantic scholar. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 84–91, New Orleans - Louisiana, June 2018. Association for Computational Linguistics.

[4] Kristjan Arumae and Fei Liu. Reinforced extractive summarization with question-focused rewards. In *Proceedings of ACL 2018, Student Research Workshop*, pages 105–111, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[5] Regina Barzilay, Kathleen R. McKeown, and Michael Elhadad. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 550–557, College Park, Maryland, USA, June 1999. Association for Computational Linguistics.

[6] Steven Bird, Robert Dale, Bonnie J. Dorr, Bryan R. Gibson, Mark Thomas Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir R. Radev, and Yee Fan Tan. The acl anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *LREC*, 2008.

[7] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022, March 2003.

[8] Lutz Bornmann and Ruediger Mutz. Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. 2014.

[9] Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. Ranking with recursive neural networks and its application to multi-document summarization. In *AAAI*, 2015.

[10] Cornelia Caragea, Jian Wu, Alina Maria Ciobanu, Kyle Williams, Juan Pablo Fernández Ramírez, Hung-Hsuan Chen, Zhaohui Wu, and Colin Giles. Citeseer x : A scholarly big dataset. In *ECIR*, 2014.

[11] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information*

Retrieval, SIGIR '98, page 335–336, New York, NY, USA, 1998. Association for Computing Machinery.

[12] Jingqiang Chen and Hai Zhuge. Automatic generation of related work through summarizing citations. *Concurr. Comput. Pract. Exp.*, 31, 2019.

[13] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[14] Yen-Chun Chen and Mohit Bansal. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[15] Jianpeng Cheng and Mirella Lapata. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, Berlin, Germany, August 2016. Association for Computational Linguistics.

[16] Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. Structural scaffolds for citation intent classification in scientific publications. In *NAACL-HLT*, 2019.

[17] Arman Cohan and Nazli Goharian. Scientific article summarization using citation-context and article's discourse structure. In *EMNLP*, 2015.

[18] Arman Cohan and Nazli Goharian. Contextualizing citations for scientific summarization using word embeddings and domain knowledge. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017.

[19] Arman Cohan and Nazli Goharian. Scientific document summarization via citation contextualization and scientific discourse. *International Journal on Digital Libraries*, 19:287–303, 2017.

[20] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. Catboost: gradient boosting with categorical features support. *ArXiv*, abs/1810.11363, 2018.

[21] Aaron Elkiss, Siwei Shen, Anthony Fader, Gunes Erkan, David J. States, and Dragomir R. Radev. Blind men and elephants: What do citation summaries tell us about a research article? *Journal of the Association for Information Science and Technology*, 59:51–62, 2008.

[22] Shai Erera, Michal Shmueli-Scheuer, Guy Feigenblat, Ora Peled Nakash, Odellia Boni, Haggai Roitman, Doron Cohen, Bar Weiner, Yosi Mass, Or Rivlin, Guy Lev, Achiya Jerbi, Jonathan Herzig, Yufang Hou, Charles Jochim, Martin Gleize, Francesca Bonin, and David Konopnicki. A summarization system for scientific documents. *ArXiv*, abs/1908.11152, 2019.

[23] Günes Erkan and Dragomir R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *ArXiv*, abs/1109.2128, 2004.

[24] Dan Gillick and Benoit Favre. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, pages 10–18, Boulder, Colorado, June 2009. Association for Computational Linguistics.

[25] Jade Goldstein-Stewart, Vibhu O. Mittal, Jaime G. Carbonnell, and Mark Kantrowitz. Multi-document summarization by sentence extraction. 2000.

[26] Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53:217–288, 2009.

[27] Vu Cong Duy Hoang and Min-Yen Kan. Towards automated related work summarization. In *COLING*, 2010.

[28] Yue Hu and Xiaojun Wan. Automatic generation of related work sections in scientific papers: An optimization approach. In *EMNLP*, 2014.

[29] Wenyi Huang, Zhaohui Wu, Chen Liang, Prasenjit Mitra, and C. Lee Giles. A neural probabilistic model for context based citation recommendation. In *AAAI*, 2015.

[30] Masaru Isonuma, Toru Fujino, Junichiro Mori, Yutaka Matsuo, and Ichiro Sakata. Extractive summarization using multi-task learning with document classification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2101–2110, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[31] Aishwarya Jadhav and Vaibhav Rajan. Extractive summarization with SWAP-NET: Sentences and words from alternating pointer networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 142–151, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[32] Kokil Jaidka, Muthu Kumar Chandrasekaran, Sajal Rustagi, and Min-Yen Kan. Overview of the cl-scisumm 2016 shared task. In *In Proceedings of Joint Workshop on Bibliometric-enhanced Information Retrieval and NLP for Digital Libraries (BIRNDL 2016)*, 2016.

[33] Kokil Jaidka, Muthu Kumar Chandrasekaran, Sajal Rustagi, and Min-Yen Kan. Overview of the cl-scisumm 2016 shared task. In *BIRNDL@JCDL*, 2016.

[34] Rahul Jha, Reed Coke, and Dragomir R. Radev. Surveyor: A system for generating coherent survey articles for scientific topics. In *AAAI*, 2015.

[35] Xiao-Jian Jiang, Xianling Mao, Bo-Si Feng, Xiaochi Wei, Bin-Bin Bian, and Heyan Huang. HSDS: an abstractive model for automatic survey generation. In Guoliang Li, Jun Yang, João Gama, Juggapong Natwichai, and Yongxin Tong, editors, *Database Systems for Advanced Applications - 24th International Conference, DASFAA 2019,*

Chiang Mai, Thailand, April 22-25, 2019, Proceedings, Part I, volume 11446 of *Lecture Notes in Computer Science*, pages 70–86. Springer, 2019.

[36] Dan Jurafsky and James H. Martin. Speech and language processing. 2019.

[37] Martin Kay. The proper place of men and machines in language translation. *Machine Translation*, 12:3–23, 1998.

[38] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. From word embeddings to document distances. In *ICML*, 2015.

[39] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *ACL 2004*, 2004.

[40] Kyle Lo, Lucy Lu Wang, Mark E Neumann, Rodney Michael Kinney, and Daniel S. Weld. S2orc: The semantic scholar open research corpus. *arXiv: Computation and Language*, 2020.

[41] Ryan McDonald. A study of global inference algorithms in multi-document summarization. In *Proceedings of the 29th European Conference on IR Research*, ECIR'07, page 557–564, Berlin, Heidelberg, 2007. Springer-Verlag.

[42] Qiaozhu Mei and ChengXiang Zhai. Generating impact-based summaries for scientific literature. In *ACL*, 2008.

[43] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[44] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[45] Sravan Nadella. Automatic text summarization using importance of sentences for email corpus, 2015.

[46] Preslav Nakov. Citances: Citation sentences for semantic analysis of bioscience text. 2004.

[47] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*, 2017.

[48] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[49] Constantin Orasan and Laura Hasler. Computer-aided summarisation - what the user really wants. In *LREC*, 2006.

[50] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[51] Chandra Prakash and Anupam Shukla. Human aided text summarizer "saar"using reinforcement learning. *2014 International Conference on Soft Computing and Machine Intelligence*, pages 83–87, 2014.

[52] Vahed Qazvinian and Dragomir R. Radev. Scientific paper summarization using citation summary networks. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 689–696, Manchester, UK, August 2008. Coling 2008 Organizing Committee.

[53] Vahed Qazvinian, Dragomir R. Radev, Saif M. Mohammad, Bonnie J. Dorr, David M. Zajic, Michael Whidby, and Taesun Moon. Generating extractive summaries of scientific paradigms. *CoRR*, abs/1402.0556, 2014.

[54] Dragomir R. Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara. The acl anthology network corpus. *Language Resources and Evaluation*, 47:919–944, 2013.

[55] John W. Ratclif and John A. Obershelp. Pattern matching: the gestalt approach, 1988.

[56] Tarek Saier and Michael Färber. unarxive: a large scholarly data set with publications' full-text, annotated in-text citations, and links to metadata. *Scientometrics*, pages 1 – 24, 2020.

[57] Dou Shen, Jian-Tao Sun, Hua Li, Qiang Yang, and Zheng Chen. Document summarization using conditional random fields. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, page 2862–2867, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[58] Zhihong Shen, Hao Ma, and Kuansan Wang. A web-scale system for scientific knowledge exploration. In *Proceedings of ACL 2018, System Demonstrations*, pages 87–92, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[59] Advaith Siddharthan and Simone Teufel. Whose idea was this, and why does it matter? attributing scientific work to citations. In *HLT-NAACL*, 2007.

[60] Juan-Manuel Torres-Moreno. *Automatic Text Summarization*. 09 2014.

[61] Chong Wang, John W. Paisley, and David M. Blei. Online variational inference for the hierarchical dirichlet process. In *AISTATS*, 2011.

[62] Jie Wang, Chengzhi Zhang, Mengying Zhang, and Sanhong Deng. Citationas: A tool of automatic survey generation based on citation content. *Journal of Data and Information Science*, 3:20 – 37, 2018.

[63] Pancheng Wang, Shasha Li, Haifang Zhou, Jin tao Tang, and Tianying Wang. Toc-rwg: Explore the combination of topic model and citation information for automatic related work generation. *IEEE Access*, 8:13043–13055, 2020.

[64] Yongzhen Wang, Xiaozhong Liu, and Zheng Gao. Neural related work summarization with a joint context-driven attention mechanism. *ArXiv*, abs/1901.09492, 2018.

[65] Yuxiang Wu and Baotian Hu. Learning to extract coherent summary via deep reinforcement learning. *CoRR*, abs/1804.07036, 2018.

[66] Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander Richard Fabbri, Irene Li, Dan Friedman, and Dragomir R. Radev. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *AAAI*, 2019.

[67] Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 452–462, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[68] Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663, Melbourne, Australia, July 2018. Association for Computational Linguistics.