ISSN 2007-9737

A Comprehensive Review on Automatic Text Summarization

Iskander Akhmetov[1,2], Sabina Nurlybayeva[2], Irina Ualiyeva[3], Alexandr Pak[1,2], Alexander Gelbukh[4]

[1] Institute of Information and Computational Technologies, Almaty, Kazakhstan

[2] Kazakh-British Technical University, Almaty, Kazakhstan

[3] Al-Farabi Kazakh National University, Almaty, Kazakhstan

[4]Instituto Politécnico Nacional (IPN), Center for Computing Research (CIC), Mexico City, Mexico

i.akhmetov@ipic.kz, a.pak@kbtu.kz, i.ualiyeva@gmail.com, gelbukh@cic.ipn.mx

Abstract. This article presents a broad overview of Automatic Text Summarization (ATS) as a downstream Natural Language Processing (NLP) task. We explore the bibliometrics, available data, methods, summary evaluation techniques, and summarization models. We start from the early methods of text summarization suggested by earlier research on the problem in the middle of the 20th century and follow the developments in the methods, approaches, and data available until recent times. We observe Artificial Neural Network (ANN) models replacing Extractive Summarization methods in favor of Abstractive ones. Finally, we compare the performance of the state-of-the-art summarization models on different datasets from various domains. And conclude that Abstractive Summarization models outperform Extractive ones based on the ROUGE score because, most of the time, "golden" or reference summaries are abstractive. However, that does not necessarily mean that Extractive summaries are bad. It only suggests that the Extractive Summary lexicon fails to match the reference summary lexicon sufficiently. Thus, we suppose there have to be other means to assess Extractive Summary quality, and at the same time, there is a need to evaluate the reference summary quality as well.

Keywords. Text summarization, natural language processing, information extraction.

1 Introduction

That is why fast information processing is a vital feature for everyone. The Text Summarization process includes plenty of challenges, even though technologies are developing, and this problem has been studied since 1958 [69]. There are two main issues that stand out:

1. Selection of essential information from a given text.
2. Representation of this information in a compressed form.

Text Summarization is a complex challenge in Natural Language Processing because it involves rigorous text semantic and lexical analysis to produce a good summary. In addition, a high-quality summary must contain salient information, be precise on the facts, and be relevant, readable, and non-redundant [108].

Researchers developed many different methods from the beginning of the research of text summarization problems. The methods differ in the number of documents they are applied to; thus, there is single and multi-document summarization.

[10] defined two classes of text summarization methods:

1. Extractive summary: includes selecting and extracting only parts of information from the original text.
2. Abstractive summary: the salient information from the original text is expressed in entirely different words.

When comparing the two methods, the second type is about conveying the salient information in an accessible form and sentences different from those appearing in the source text. In contrast, the extractive method assembles a summary from the source text, finding the most important sentences. Thus, extractive summaries are easier to get and are expected to yield better results than abstractive summaries [25]. On the other hand, the second task is more difficult because it involves complex techniques like Natural Language generation [16].

Nowadays, researchers focus on Abstractive Summarization methods [43]. Nevertheless, Extractive Summarization is still in trend, as seen from the research papers in the last two years [54, 90, 105].

In addition to the complexity of forming a summary, an open question in the scientific community is its quality assessment. The quality metric of summaries should consider not only the number of words shared by the candidate summary and reference summary but also the informativeness of the candidate summary concerning the source text.

Other review articles cover only distinct aspects of the Automatic Text Summarization problem. For example, they covered the approaches and techniques [5, 76], the methods [89], summary evaluations techniques [96]. The fragmentation of the topics of the reviews makes it difficult for researchers, especially those who are just starting to study this area. Much work is required, and it may take a lot of work to conduct a thorough analysis.

This article aims to outlay a broad overview of the text summarization problem, including existing datasets, methods for summarizing text, and methods for assessing the quality of automatic summations.

Our contribution to the scientific body of knowledge is in:

1. Gathering a wide range of aspects related to automatic text summarization in one place.
2. Carrying out a comparative analysis of the performance of different models on various datasets from diverse domains.
3. As a result of reviewing evaluation metrics used today, states the need for a better metric for summary quality assessment than the currently used ROUGE metric, which is an industry-standard nowadays.

In the following sections of this paper, we provided a Bibliometric review of the “text summarization” topic, described the Data and Methods commonly used, showed the Evaluation Metrics landscape, compared the most famous text Summarization Models, and got to the Results and Conclusions.

2 Bibliometrics

We used the Scopus database to analyze text summarization publications bibliometrically. Scopus is one of the largest bibliometrics databases of peer-reviewed literature that covers a wide range of subjects. Scopus was inspired by the bird Hammerkop (Scopus umbrella), which reportedly has excellent navigation skills. The entire Scopus database goes back to 1966. At the end of 2021, the collection contained over 40,000 titles from approximately 11,678 international publishers, of which nearly 35,000 journals are peer-reviewed in top-level subject fields. Scopus covers various publication formats (books, journals, conference proceedings, and others) in science, engineering, medicine, social sciences, and arts and humanities. In addition, Scopus includes content from various platforms, both free and required subscriptions, such as Google Scholar.

Web of Science (WoS) and Scopus databases have a high association, i.e., overlap in journal

indexing but index different journals. However, Scopus offers more coverage than WoS. Compared with Google Scholar, Google Scholar can retrieve even the most obscure information and is not limited to recent articles, but Scopus offers results with more consistent accuracy. Datasets from Scopus were downloaded and processed on the Python program with Jupyter Notebook.

We created a search query "Extractive Summarization OR Abstractive Summarization OR Summarization OR Text Summarization" (further, we will call this query "Text Summarization"), and Scopus returned more than 57,000 papers.

According to the Scopus database, almost 7000 authors from 160 countries and 160 institutions published their research on the Text Summarization area in 29 107 conferences Table 1.

Table 1. Scopus dataset properties

Papers	57 255
Authors	6654
Institutions	160
Source	128
Conferences	29 107
Subject area	28
Countries	160

Table 2 shows the institutions (from 1958 to 2021) ordered by several publications in the Text Summarization area. Another meaninigful information is about World University Rankings. Times Higher Education (THE) methodology groups University Ranking 13 metrics into five major dimensions: Research (30%), Citations (30%), Teaching (30%), International outlook (7.5 %), Industry income (2.5%). THE describes research impact as an indicator of a university's role in distributing new scientific knowledge and ideas.

Among these top 10 institutions, Columbia University is ranked seventh in the World University Ranking list; six are ranked in the top 200, and 50% of the institutes are located in China.

Table 3 shows the ten most productive journals on Summarization research. A noticeable leadership in the number of publications for 2020-2021 belongs to the Journal of Lecture Notes in Computer Science (LNCS), including its subseries Lecture Notes in Artificial Intelligence (LNAI) And Lecture Notes in Bioinformatics (LNBI).

We exported citation, bibliographical, and author keywords information of 1289 articles from VosViewer Application [104]. An analysis of the Scopus dataset was carried out in the context of the topics of publications and using search queries to identify the most common keywords in text summarization articles.

Data in Table 4 shows the number of articles in which the Summarization keyword is paired with other keywords. Table 4 is formed from the top of the most frequently encountered keywords. For example, the intersection shows the number of articles from which it can be seen that the keyword 'Natural Language Processing systems' is often used in conjunction with other keywords.

This analysis found that the Text Summarization Topic is popular for Natural Language Processing and Computational Linguistics, as it is often paired with these keywords.

The top 10 most prolific authors can be identified according to Scopus. For example, data in Table 6 shows the list of authors with the most significant number of articles on Text Summarization. The leader is Lloret Elena, cited in 459 articles, and 160 citations belong to the review article [68].

Table 7 shows the most valuable authors for the Summarization area. The primary influence is the number of publication citations from text summarization. By statistics, the most valuable publication is done by Liu Bing and Hu Minqing.

In the co-authorship analysis, we included 113 countries affiliated with 2967 authors.

Data in Table 8 shows the co-authorship matrix, the number of joint publications at the intersection. There are eight central communities within the general distribution, where authors have joint publications.

The year of publication is another critical piece of information for bibliometric research. Our first task was to explore the years of publication in the

[1] Total Publications (2017-2020)
[2] Total Citations
[3] Cite Score 2020
[4] Publications on Text Summarization (2000-2021)
[5] In 2019.

Table 2. Top institutions by the number of publications according to Scopus database

Rank	Institution	World University Rankings 2021-2022	Total Publications	Country
1	Chinese Academy of Sciences	73	110	China
2	Peking University	59	79	China
3	Carnegie Mellon University	85	62	United States
4	University of Chinese Academy of Sciences	73	61	China
5	Columbia University	7	55	United States
6	Universitat d'Alacant	845	47	Spain
7	Hong Kong Polytechnic University	267	44	Chine
8	Beijing University of Posts and Telecommunications	737	42	China
9	The University of Sheffield	150	41	United Kingdom
10	Université d'Avignon et des Pays du Vaucluse	1493	41	France

Table 3. The top 10 most productive journals by the number of publications on text summarization

	Journal	Publ.[1]	Citations[2]	CS 2020[3]	TS Publ.[4]
1	LNCS, LNAI, LNBI	82 766	141 179	1.8	139
2	Advances In Intelligent Systems And Computing	29 624	26 852	0.9[5]	42
3	CEUR Workshop Proceedings	18 904	15 553	0.8	42
4	Communications In Computer And Information Science	19 615	15 364	0.8	23
5	Expert Systems With Applications	2710	34 460	12.7	21
6	ACM International Conference Proceeding Series	31 048	35 869	1.2	19
7	IEEE Access	41 670	201 619	4.8	19
8	Information Processing And Management	541	4676	8.6	16
9	International Conference on Recent Advances In Natural Language Processing (RANLP)	267	516	1.9	16
10	Procedia Computer Science	8236	24 640	3.0	15

summarization field. Fig. 1 shows a histogram of the publication year. In total, 57 255 documents we identified beginning from 1958 (the year of first publication by Luhn).

We show that the number of publications grows steadily and has significantly increased from 1995 to 2015, explained by the fact that publications of papers on Machine Learning and Neural Network methods firstly applied to summarization by [58] and [93] respectively. The highest number of publications on Text Summarization was 6608 reported in 2019; see1.

Investigating how much other researchers from different generations reference methods employed for text summarization since 1958, surprisingly, we find that the method introduced by Luhn [69] at the beginning of this period is still referenced. Moreover, the number of references has been steadily growing since 1997 up to now, from 42 to 217 in 2019, reaching the total number of 3681 references in 62 years, which might mean that the method of sentence/word weighing is a fundamental end efficient; see Fig. 2. However, after 2020, the trend declined, reaching a point with

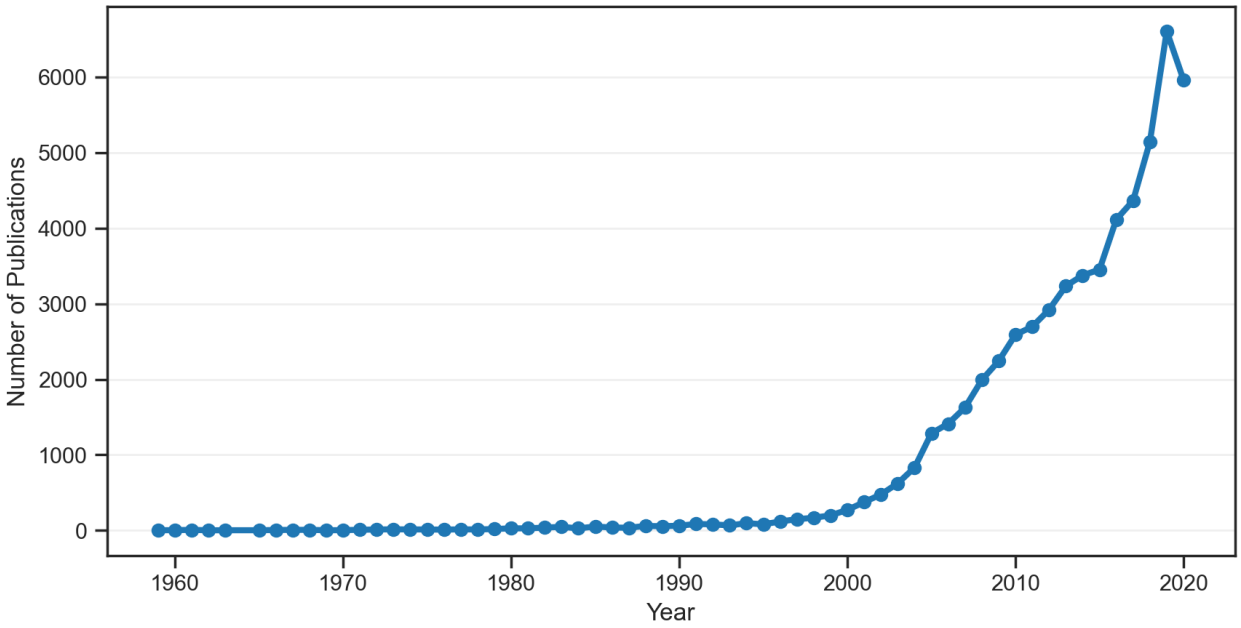


Fig. 1. Documents by year

Table 4. Top 10 most co-occurred keywords with text summarization

	Keyword	Occurrence
1	Natural Language Processing systems	1887
2	Text Processing	1871
3	Text Summarization	1812
4	Semantics	1025
5	Information Retrieval	708
6	Computational Linguistics	666
7	Data Mining	635
8	Automatic Text Summarization	499
9	Natural Language Processing	488
10	Artificial Intelligence	474

an indicator of 61 and 81 in 2021.

Similar popularity is expressed for more recent papers on the Machine Learning method first applied by [58], [72] and [93] in 2015; see Fig. 1. [93] devised an attention-based summarization approach (ABS) for a sentence-level summary generation; [58] classified the sentences as summary sentences with a naive-Bayes classifier.

Fast Reading Understanding and Memory Program (FRUMP) [20] and hidden Markov models (HMM) [19] models are rarely mentioned in scientific papers; see Fig. 3.

Table 9 shows the most cited articles in the Extractive Text Summarization research area according to the Scopus database. It is noticeable that the most cited publications were before 2000, but LexRank entered the top 2 articles and is the newest from the proposed list.

Table 10 shows similar statistics, but only for Abstractive Summarization. The top 10 most cited papers included articles published after 2015, and the top 5 papers included an article from 2020 with the proposed Bottom-up abstractive summarization method.

3 Data

The amount of data available for experiments in text summarization remained low with several datasets of not more than 1000 articles and their summaries until 2003 when the Gigaword

Table 5. Keywords co-occurrence

	Natural Language Processing systems	Text Summarization	Text Processing	Semantics	Natural Language Processing	Artificial Intelligence	Computational Linguistics	Information Retrieval	Data Mining
Natural Language Processing systems	-	1887	5681	13 414	15 469	7444	15 463	7321	7461
Text Summarization	1887	-	1812	1025	750	474	666	708	635
Text Processing	5681	1812	-	10 923	9198	6151	8209	7680	9821
Semantics	14 600	1025	10 923	-	3753	11 852	11 884	10 867	8878
Natural Language Processing	15 469	488	9198	3753	-	9006	15 867	7901	8175
Artificial Intelligence	7444	474	6251	11 852	9006	-	4212	7275	16 245
Computational Linguistics	15 463	666	8209	11 884	15 867	4212	-	3567	2487
Information Retrieval	7321	708	7680	10 867	7901	7275	3567	-	12 125
Data mining	7478	635	9821	8878	8175	16 245	2487	12 125	-

Table 6. The top 10 prolific authors in text summarization research area. Note: Number of articles on Text Summarization (TSP), Total Publications (TP), h-index is Hirsch-Index, Total Citations (TC), Country ISO 3166 code

	Author	TSP	TP	h-index	TC	Current affiliation	Country
1	Lloret Elena	37	452	12	547	Universitat d'Alacant	ES
2	Salim Naomie	31	219	24	2559	Universiti Teknologi Malaysia	MY
3	Saggion Horacio	30	153	23	2055	Universitat Pompeu Fabra Barcelona,	ES
4	Lins Rafael Dueire	25	84	13	681	Universidade Federal Rural de Pernambuco,	BR
5	Palomar Manuel	21	867	17	1013	Universitat d'Alacant	ES
6	Gupta Vishal	14	48	12	1376	University Institute of Engineering and Technology	IN
7	Abujar Sheikh	12	54	8	177	Independent University, Bangladesh	BD
8	Hossain Syed Akhter	12	95	9	308	University of Liberal Arts Bangladesh	BD
9	Kallimani Jagadish S.	9	43	6	100	Visvesvaraya Technological University,	IN
10	Alami Nabil	7	8	4	69	Faculté des Sciences Dhar El Mahraz, Université Sidi Mohamed Ben Abdellah,	MA

dataset with almost 4M article-summary pairs was introduced [38].

The availability of large datasets in the mid-2000s [47] opened new opportunities for applying a wide variety of algorithms, methods, and approaches, including Deep Learning, which became a new brand name for neural network algorithms, known since the 1980s.

The growth of computing power at the beginning of the 2000s and dataset availability gave an immediate boost in the number of publications regarding text summarization, which continued to grow exponentially supported by the appearance of new large datasets; see Fig. 4.

Comparing the total number of scientific publications with those, especially text summarization, we can see their exponential growths correlate. Technological advancements in computer hardware may have affected the narrow area of NLP

Table 7. The top 10 most valuable authors on the topic of Text Summarization by the number of citations

Author	The most cited article in Summarization research area	TSP	TSC	Current affiliation	Country
Liu Bing	Mining and summarizing customer reviews [49]	5	7302	The University of Illinois at Chicago	US
Hu Minqing	Mining and summarizing customer reviews [49]	3	5565	MySpace Inc.	US
Radev Dragomir	LexRank: Graph-based lexical centrality as salience in text summarization [26]	18	4899	Yale University	US
Erkan Gunes	LexRank: Graph-based lexical centrality as salience in text summarization [26]	3	1918	University of Michigan	US
Zhai Chengxiang	Topic sentiment mixture: Modeling facets and opinions in weblogs [71]	11	1789	University of Illinois Urbana-Champaign	US
Liu Pengfei	Searching for effective neural extractive summarization: What works and what's next [116]	3	1264	Carnegie Mellon University	US
Lu Yue	Latent aspect rating analysis on review text data: A rating regression approach [109]	8	1249	The Nanjing University of Aeronautics and Astronautics	CHN
Li Wei	Pachinko allocation: DAG-structured mixture models of topic correlations [62]	30	1209	Yahoo Research Labs	US
Liu Peter. J.	Get to the point: Summarization with pointer-generator networks [97]	4	1203	Google LLC	US
McKeown Kathleen	Sentence fusion for multi-document news summarization [6]	9	1084	Columbia University	US

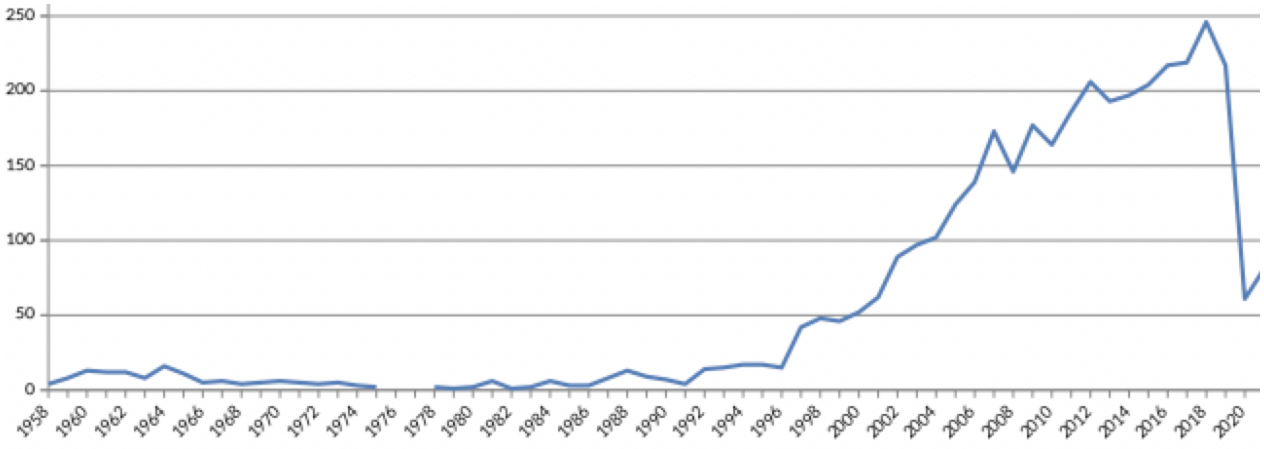


Fig. 2. Words/Sentence weighing (reference count)

- text summarization encompassing all science branches.

Although scientific publications are growing, the most common data are from News Datasets. The statistics on the available categories and data are presented in the Table 11. We see that News Data shares leadership with Scientific Papers. The inferior documents comprise a small fraction of the total and are displayed in a separate category called Other.

Table 8. Co-authorship matrix

	Abujar S.	Hossain S.A.	Masum A.K.M.	Alami N.	Meknassi M.	Chen J.	Wang X.	Yu H.	Chen Q.	Chen X.	Li P.	Wang H.	Zhang C.	Ferreira R.	Freitas F.	Lins R. D.	Simske S. J.	Lloret E.	Palomar M.	Saggion H.	Vodolazova T.	Wang J.	Yang Z.	Zhang Y.	Zhang L.	Zhang X.	Wang Y.	Zhang H.	
Abujar S.	-	8	6																										
Hossain S.A.	8	-	5																										
Masum A.K.M.	6	5	-																										
Alami N.				-	5																								
Meknassi M.				5	-																								
Chen J.						-	1	1																	1				
Wang X.						1	-		1																1			1	1
Yu H.						1		-																	1				
Chen Q.							1		-																				
Chen X.										-	1	1	1	1															
Li P.										1	-		1																
Liu X.										1		-		1															
Wang H.										1	1		-																
Zhang C.										1		1		-												1	1		
Ferreira R.															-	4	4	4											
Freitas F.															4	-	5	5											
Lins R. D.															4	5	-	5											
Simske S. J.															4	5	5	-											
Lloret E.																			-	11	1	4							
Palomar M.																			11	-	1	2							
Saggion H.																			1	1	-								
Vodolazova T.																			4	2		-							
Wang J.																							-	3					1
Yang Z.																							3	-					
Zhang Y.						1	1	1																	-	1		1	
Zhang L.														1											1	-			
Zhang X.														1													-		
Wang Y.							1																		1			-	
Zhang H.							1																1						-

3.1 Scientific Datasets

The scientific datasets section consists of scientific publications and articles. The dataset structure contains the author's name, article, and annotation, which allows the use of the data to summarize the text. The following datasets can be attributed to the scientific category: arXiv, PubMed, and BigPatent. Scientific summarization datasets properties are shown in Table 12.

3.1.1 arXiv

For almost 30 years, arXiv[6] is serving the scientific research community by providing access to scientific articles, from the various branches of math, physics, subdisciplines of computer science, and everything in between and around, including statistics, electrical and mechanical engineering, bioinformatics, and economics.

ArXiv dataset for long document summarization contained 215K documents from the official database website and was first collected by [17].

3.1.2 PubMed

In 1996, PubMed made it possible to access more than 28 million links to biomedical and life articles from the MEDLINE database. Broad free access to the PubMed system appeared in June 1997.

The PubMed dataset comprises 133K scientific publications from the PubMed database [17].

[6] arXiv is a free open-access archive for 1 975 103 scientific articles in the fields of physics, computer science, mathematics, statistics, electrical engineering, quantitative biology, quantitative finance, systems science, and economics (`https://arxiv.org/`).

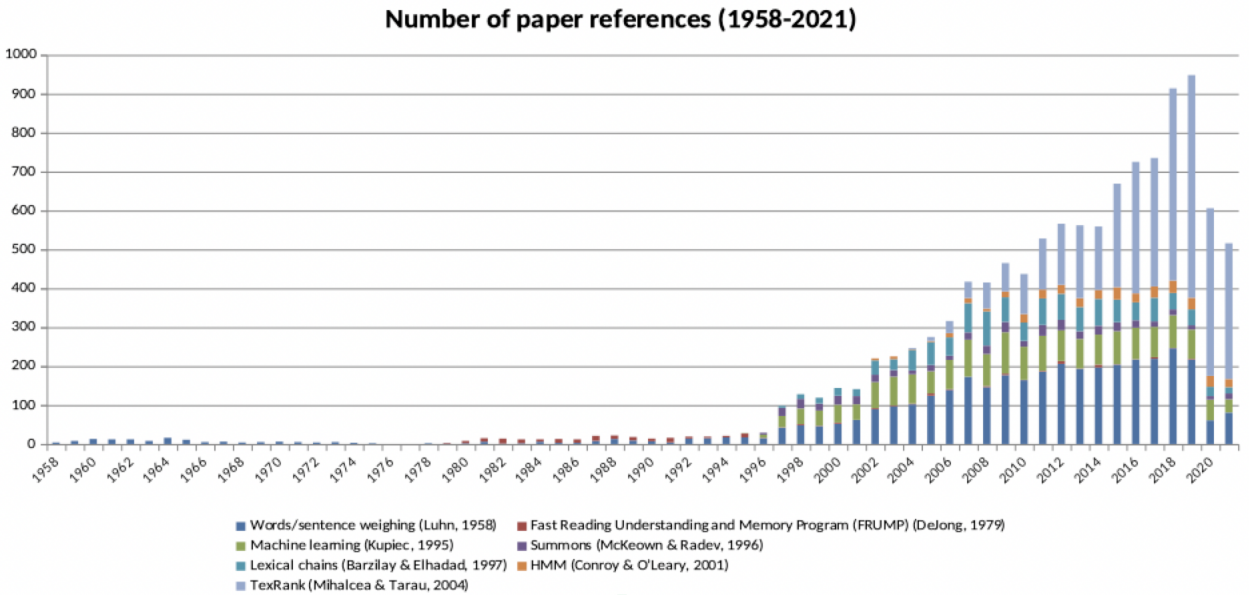


Fig. 3. Other summarization methods (reference count)

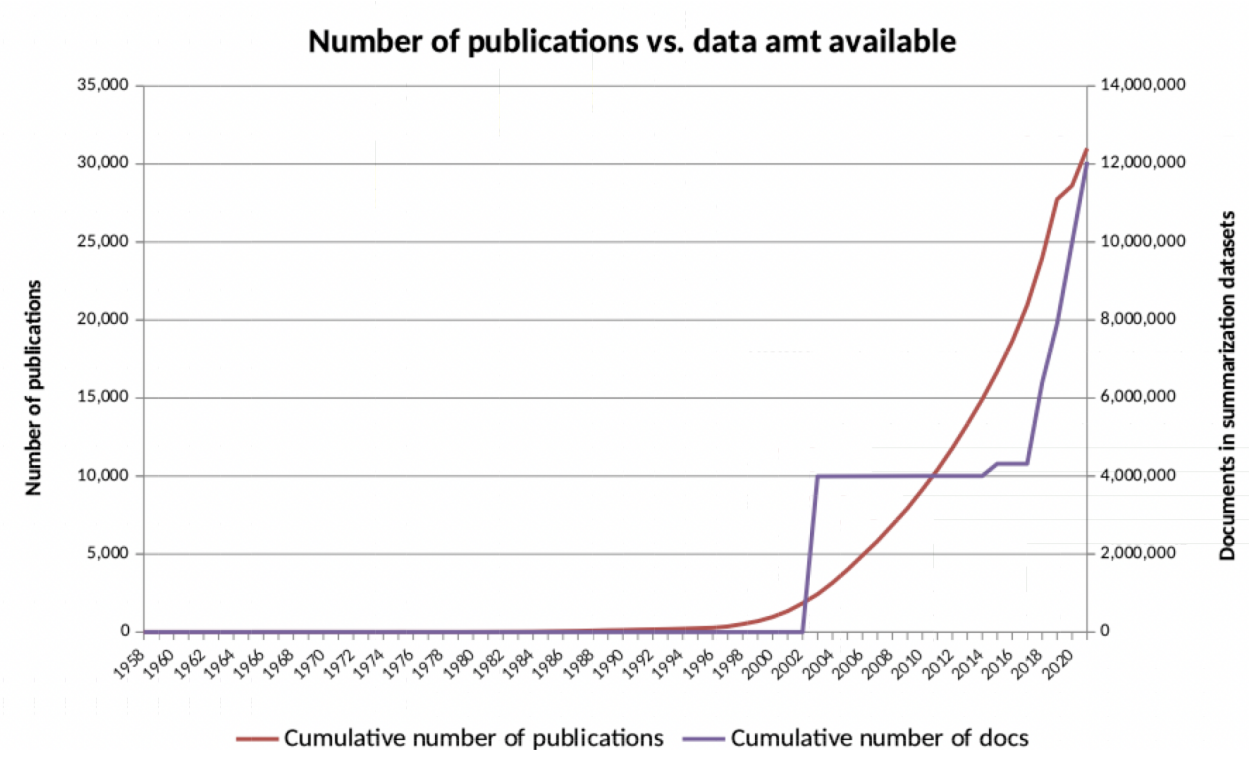


Fig. 4. Number of publications vs. data amount available

Table 9. The most cited articles in extractive text summarization research area

	Title	Authors	Citations	Year
1	Use of MMR, diversity-based reranking for reordering documents and producing summaries [14]	Carbonell J., Goldstein J.	1736	1998
2	LexRank: Graph-based lexical centrality as salience in text summarization [26]	Erkan G., Radev D.R.	1633	2004
3	Trainable document summarizer [59]	Kupiec J., Pedersen J., Chen F.	775	1995
4	TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages [46]	Hearst M.A.	774	1997
5	Movie review mining and summarization [118]	Zhuang L., Jing F., Zhu X.-Y.	635	2006
6	Learning algorithms for keyphrase extraction [103]	Turney P.D.	600	2000
7	Generic text summarization using relevance measure and latent semantic analysis [37]	Gong Y., Liu X.	560	2001
8	Incorporating copying mechanism in sequence-to-sequence learning [41]	Gu J., Lu Z., Li H., Li V.O.K.	522	2016
9	Deriving concept hierarchies from text [94]	Sanderson M., Croft B.	431	1999
10	Summarizing text documents: Sentence selection and evaluation metrics [36]	Goldstein J., Kantrowitz M., Mittal V., Carbonell J.	323	1999

3.1.3 BigPatent

BigPatent is a dataset of 1.3 million US patent documents. It contains patents filed after 1971 in nine different technological fields. For the summarization problem, the patent abstract is treated as the golden summary and its description as the source text [98].

BigPatent has the following properties compared to other summarization datasets:

— Summaries contain a more dense discourse structure with more recurring entities.

— Salient information is distributed evenly in the input text.

— Shorter extractive fragments are in the summaries.

3.2 News Datasets

News is the largest category in document content, as described in Table 11.

The category consists of the following datasets: CNN, Daily Mail, Gigaword, X-Sum, Newsroom, and datasets from DUC and TAC conferences. News summarization datasets properties are shown in Table 13.

Table 10. The most cited articles in abstractive text summarization research area

	Title	Authors	Citations	Year
1	Get to the point: Summarization with pointer-generator networks [97]	See A., Liu P.J., Manning C.D.	955	2017
2	A neural attention model for sentence summarization [92]	Rush A.M., Chopra S., Weston J.	759	2015
3	Abstractive text summarization using sequence-to-sequence RNNs and beyond [74]	Nallapati R., Zhou B., dos Santos C., Gulçehre Ç., Xiang B.	597	2016
4	Abstractive document summarization with a graph-based attentional neural model [101]	Tan J., Wan X., Xiao J.	158	2017
5	Bottom-up abstractive summarization [32]	Gehrmann S., Deng Y., Rush A.M.	146	2020
6	Deep communicating agents for abstractive summarization [15]	Celikyilmaz A., Bosselut A., He X., Choi Y.	114	2018
7	Toward abstractive summarization using semantic representations [65]	Liu F., Flanigan J., Thomson S., Sadeh N., Smith N.A.	106	2015
8	Deep recurrent generative decoder for abstractive text summarization [61]	Li P., Lam W., Bing L., Wang Z.	85	2017
9	Abstractive text summarization using LSTM-CNN based deep learning [100]	Song S., Huang H., Ruan T.	76	2019
10	A framework for multi-document abstractive summarization based on semantic role labelling [53]	Khan A., Salim N., Jaya Kumar Y.	75	2015

3.2.1 Document Understanding Conference (DUC)

DUC has been run annually from 2001 until 2008 and has been a significant forum for comparing summarization systems on a shared test set[7].

DUC 2001-2004 datasets are more related to multi-documental summarization, and DUC 2005-2007 datasets are also related to this topic but are query-focused.

The DUC datasets are news data contained in three datasets related to the conference year, divided into various thematic clusters.

Note that each cluster includes 2-4 summaries composed by professional experts [44].

[7]Online proceedings of the conferences are available at https://duc.nist.gov/data.html

Table 11. Summarization datasets amount of documents by domain area

Dataset topic	Amt. of docs
Emails	18 000
Instructions	200 000
Legislation	23 000
News	5 897 122
Science	1 648 250
Short story	120 000
Total	7 906 372

3.2.2 Text Analysis Conference (TAC)

The continuation of the DUC conference was the TAC competition, created to study the field of natural language processing. Each participant received a test dataset and result assessment [8].

TAC 2010 is a popular summation dataset that collects from 440 documents. The dataset can be divided into five main categories: Accidents and natural disasters, Terrorist attacks, Investigations and Litigation, Health and safety, and Disappearing resources.

3.2.3 Gigaword

The Gigaword summarization dataset was introduced by Graff et al. in 2003 [38] and consisted of 8.6 mln short news articles for the headline generation or one-sentence summary generation task. The actual Gigaword dataset was presented by [92]. The dataset comprises 3.8M training, 189k development, and 1951 test documents.

3.2.4 CNN and Daily Mail

The CNN / Daily Mail dataset [74] has been used for summary evaluation. It is called "anonymized" since it uses tags instead of the named entity.

Human-generated abstractive summary highlights comprised news stories on CNN and Daily Mail as questions and reports as the corresponding passages.

[8]The online proceedings of the conferences are available at `http://tac.nist.gov/2010/`

CNN CNN abstractive summary dataset consists of 92,000 documents generated from the CNN website and first used in 2016 [74].

Daily mail The Daily Mail abstractive summary dataset consists of 219,000 documents generated from the Daily Mail website, first used in 2016 [74].

3.2.5 Extreme Summarization (X-Sum)

X-Sum [75] is a dataset that does not suit extractive summarization and encourages an abstractive summarization approach. The dataset task is to create a short, one-sentence news summary for a news story provided as input. Data was collected by scraping online article pages from the BBC website. The dataset contains 204K training, 11K validation, and 11K test sample sets. The article's average length is 431 words (20 sentences), and the summary length is 23 words.

3.2.6 Cornell Newsroom

Cornell Newsroom [40] is a massive dataset for summarization systems training and evaluation. The dataset consists of 1.3M articles and summaries composed by authors and editors from 38 significant newsroom sources. The summaries are collected from search and social metadata between 1998 and 2017 and use various summarization strategies combining extraction and abstraction.

3.2.7 NY Times Corpus

The New York Times Annotated Corpus consists of over 1.8M articles published between 01.01.1987 and 19.06.2007, augmented with article metadata [95].

The data set consists of 650K article-summary pairs, and library researchers created most of the article summaries manually. Also, over 1.5M documents have at least one tag, such as topics, places, persons, organizations, and titles.

Table 12. Scientific summarization datasets properties

Dataset	Num. docs	Avg. words/article	Avg. words/summ.
arXiv	215 913	4938.0	220.0
PubMed	133 215	3016.0	203.0
BigPatent	1 341 362	116.5	3572.8

Table 13. News summarization datasets properties

Dataset	Num. docs	Avg. words/article	Avg. words/summ.
CNN/Daily Mail	312 084	781.0	56.0
BBC News	2225	-	-
Gigaword	3 990 951	31.4	8.3
X-Sum	226 711	431.0	23.0
Cornell Newsroom	1 321 995	658.6	26.7
NY Times Corpus	650 000	530.0	38.0
DUC-2001	309	100.0	-
DUC-2002	567	100.0	-
DUC-2004	500	-	-
TAC-2014	220	-	235.6

3.2.8 BBC News

The BBC News dataset of 2225 classified articles stemmed from BBC News in 2004 and 2005 labeled in business, entertainment, politics, sports, and technology[9]. The dataset is made freely available for non-commercial and research purposes only, and all data is provided in pre-processed format [39].

3.3 Books

The task of summarizing the texts of books is no less urgent; for its solution, there is another category of datasets - books. Popular book datasets' statistics are shown in Table 14.

Table 14. Books datasets properties

Dataset	Num. docs	Mean doc. size (KiB)	Size (GiB)
Bookcorpus	11 038	419.37	4.63
BookCorpusOpen	17 868	369.87	6.30
Books3	197 000	538.36	100.96

[9]All rights, including copyright, in the content of the original articles, are owned by the BBC. `http://mlg.ucd.ie/datasets/bbc.html`

3.3.1 Bookcorpus

The dataset compilers collected two datasets, one consisting of films and annotations, and the second is a BookCorpus Dataset [117].

BookCorpus dataset consists of 11K books taken from the site with electronic books. It is important to note that this dataset does not apply to copyright, as it is only found in free books by unpublished authors. Also, to preserve the purity of the experiment, the researchers left only books with more than 20,000 words in 16 different genres in the dataset.

3.3.2 Books1 or BookCorpusOpen

BookCorpusOpen is an expanded version of the BookCorpus [117]. However, due to the BookCorpus dataset availability issue and the possibility of collecting a more extensive version, the second version of BookCorpus was collected by enthusiasts. This version has 17.9K books containing two fields: title and unprocessed book text. The structure and available amount of data in the corpus is similar to the corpus named Books1 used in the development of GPT-3 [13] by OpenAI[10].

[10]OpenAI is an Artificial Intelligence (AI) research and development company supported by Elon Musk. The

3.3.3 Books3 or Bibliotik

Books3 is a corpus of books taken from a sample of the Bibliotik. This dataset is Shawn Presser's work and is part of The Pile dataset [31, 86].

Bibliotik contains fiction and nonfiction books and is more extensive than BookCorpusOpen. It includes all of the documents in plain text format, with around 197,000 readers that were processed similarly to BookCorpus. The corpus's structure and available amount of data are similar to Books2.

3.4 Other Datasets

Another separate category consists of datasets that contain unclassified information inside - the contents of checks, a set of texts from the Wiki. Statistics for datasets are described in Table 15.

Table 15. Other summarization datasets properties

Dataset	Num. docs	Avg. words/article	Avg. words/summ.
Billsum	22 218	1533.0	500.0
WikiHow	230 843	579.8	62.1
WikiLingua	42 783	391.0	39.0

3.4.1 Billsum

The BillSum dataset consists of US training bills and test bills. The bills were collected from the US Government Publishing Office (GPO) Govinfo service [55].

In total, there are 22.3K bills from sessions of the US Congress in the dataset, which were collected from 1993 to 2018. The California Legislative Counsel has prepared summaries for bills since 2015-2016.

organization's mission is to ensure that AI generally benefits all of humanity.

3.4.2 WikiLingua

WikiLingua is a large-scale, multilingual dataset to evaluate cross-lingual abstractive summarization models. It consists of texts in 18 languages extracted from WikiHow (pairs of the article and its summary). The written human texts of WikiHow are high-quality textual data on various topics. The golden standard for the summarization principle is mapping texts across languages with the same image occurrences feature. Briefly, the dataset consists of 141.5K unique English articles. Each of the other 17 languages has, on average, 42.8K articles that align with an article in English.

3.4.3 WikiHow

WikiHow is a dataset consisting of more than 200k long-sequence pairs. Each document is collected from the online knowledge base of the same name by combining paragraphs in the article and identifying generalizing sentences [56].

4 Summarization Methods

To provide the big picture of the methods landscape from the beginning, we provide the classification of text summarization methods; see Fig. 5.

Firstly, we need to classify the approaches used in text summarization:

1. **Extractive Summarization**

   The Extractive Summarization methods select informative sentences from the source document based on specific criteria to construct a summary. In other words, such methods cut off unnecessary sentences by some informative measure. Extractive Summarization's main challenge is choosing the significant sentences from the input document as in summary. There are several approaches to measure sentence informativeness, e.g., the statistical one based on the frequencies of significant and auxiliary words [29]. A reader can find them further in the subsection 6.1. Another approach is the usage of language models or as they are often called in the

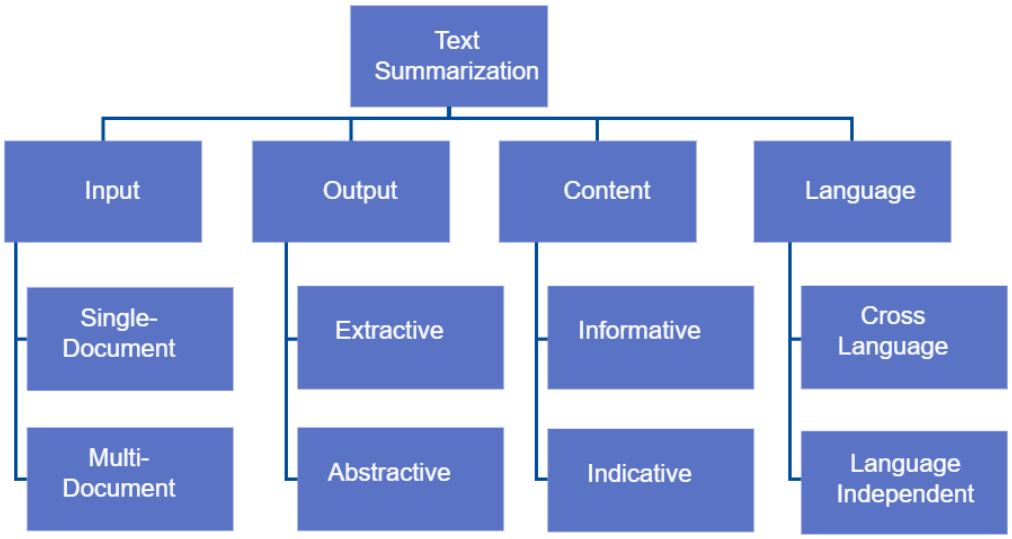


Fig. 5. Text summarization methods classification [1, 87]

scientific literature on the word embeddings [12, 22, 73, 85], these methods are presented in the subsection 6.2. Additionally, some traditional methods have evolved due to applying word embeddings to their processing, which readers can find further in the concrete subsections for these methods.

2. **Abstractive Summarization**

   Abstractive Summarization methods generate summaries by constructing new short sentences, like a human being. The summary may contain phrases that are not present in the original text. Generation and compression techniques are needed to create an abstract summary language. Abstract text summarization has two approaches: 1) structure-based and 2) semantics-based approaches. Both of them use ideas of word embeddings; in other words, inside of their processing pipelines, there are one of the words embedding models [4,12,13,22,73,85]. This will be described in more detail in the relevant subsections under section 6.3.

3. **Informative Summarization** An informative summary represents the original document in full. Therefore, it contains all the salient information necessary to convey the core meaning of the source text and omits ancillary information.

4. **Indicative Summarization** An indicative summary's primary purpose is to recommend the article's contents without giving details on the article's content. It can serve as a teaser to motivate the user to retrieve the full text. Examples of indicative summaries include book annotations, web search result snippets, and movie trailers.

5. **Single Document** Single document summarizers aim to summarize one single document.

6. **Multi-document Summarization** Multi-document summarizers as a source use a collection of documents related to a common subject or event and produce the summary on multiple documents in the temporal order. For example, it can be used in the literature review process of scientific work or in compiling a subject report article to receive short and concise information on a subject, reducing redundancy [3]. The systems can be as simple as picking the most crucial document and using it for a single-document summarizer [113], or use ontologies and focusing on query [30, 84].

Alternatively, summarize all documents individually, merge the summary, and then summarize the merged sum.

7. **Language-Independent Text Summarization** Language-Independent Text Summarization is the process of multilingual text summarization.

8. **Cross-Language Text Summarization (CLTS)** CLTS is defined as examining the document in a language to learn the prominent factors, generating a short, suitable, and accurate document summary in a specific language [70]. Nowadays, the systems have employed compressive and abstractive frameworks to maximize summaries' usefulness and grammatical supremacy. However, these models need unique resources for a language and unification of different models, limiting their applicability in summary generation in various languages [70].

5 Evaluation Metrics for Quality Evaluation of Summaries

Summary quality evaluation score is a critical factor affecting the success of generalization tasks. Currently, most existing methods measure the similarity of the generated abstract with the gold standard written by people. In this section, the metrics are arranged in order of publication year.

5.1 Human Evaluation

The problem of evaluating a summary of a text is not a trivial task since the multidimensionality of the semantic space in which the main characteristics of the generated text are calculated can potentially have an infinite number of estimates and their interpretations [34, 36, 80]. Nevertheless, in our opinion, the following crucial points can be distinguished:

1. Manual labeling can be redundant on the one hand and insufficient on the other; therefore, its reuse in related tasks requires additional manual labor.
2. There are different circumstances of manual labeling, namely subjectivity, and conflicts between assessors. On average, the share of agreement between assessors can achieve 70 percent. Such a feature complicates the development of big data sets [33].
3. One of the reasons for using manual labor to assess the abstractive summarization is that there is a single gold standard for the target variable in the training dataset. It contradicts the very nature of multiple representations of meaning in natural languages. Thus, metrics and scores based on word coincidence are poorly suited to abstractive summarization.
4. The main characteristics measured using automatic metrics are precision and recall in terms of similarity to a golden standard; in other words, one can call it topic coverage and text redundancy. There is an interest in plenty of text metrics in the academic community and business, such as readability, coherence, informativeness, conciseness, etc. Note that there is a scientific gap in computational linguistics and natural language processing associated mainly with the psycholinguistic nature of the perception of short abstracts. Depending on the task, there are also such assessments as artistry, commitment, objectivity, etc.

5.2 BiLingual Evaluation Understudy (BLEU)

BLEU metric is designed for the automated assessment of machine translation, and its behavior correlates well with the human assessment [81]. Therefore, it is widely used in machine translation systems. Moreover, it has been adapted for the problem of assessing the quality of a text summary [64].

The main idea behind BLEU is to measure the proximity between a generated translation and a set of gold standards. The closeness is calculated based on the weighted average of the variable-length n-gram matches between generated and targeted human translations.

Numerical experiments have shown that the weighted average, the BLEU, is highly correlated

with the estimates made by people. Likewise, the authors of [64] used BLEU to evaluate the results of quasi-referencing, guided by the consideration that the closer the generated resume to the gold standard in terms of n-grams, the better the generative language model performed.

The idea of the metric is very close to ROUGE, which also estimates the proximity between texts using n-grams; the difference between the metrics is the normalizing factor. Later, both metrics were combined into one metric through the geometric mean.

BLEU is a metric based on precision. The Brevity Penalty (BP) is introduced to emulate recall as compensation for the possibility of too-short translations with a high precision score.

The BP calculation formula is given in Equation (1):

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-\frac{r}{c})} & \text{if } c \leq r, \end{cases}, \tag{1}$$

where c and r refer to the length of the hypothesis and the reference translations.

The resulting BLEU score calculation is as follows in Equation (2):

$$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right), \tag{2}$$

where n refers to the orders of n-gram considered for p_n and w_n refers to the weights assigned for the n-gram precision. For more details, calculations of p_n is described below in Equation (3):

$$p_n = \frac{\sum_{C \epsilon \{\text{Candidates}\}} \sum_{n-gram \epsilon C} count_{clip}(n-gram)}{\sum_{C' \epsilon \{\text{Candidates}\}} \sum_{n-gram' \epsilon C'} count(n-gram')}. \tag{3}$$

One of the most critical constraints of the BLEU metrics is that it is based on the assumption that it needs to match human judgment on average on an extensive test corpus because scores on individual sentences will often vary from human judgments.

5.3 BERT Score

One of the most correlated human evaluation metrics is BERT Score [115].

BERT Score uses contextualized token embeddings of a pre-trained BERT model. It calculates the semantic proximity of two sentences by summing the cosine proximity between their token embeddings.

The process consists of the following steps:

1. Contextual embedding;
2. Pairwise cosine similarity;
3. Maximum similarity;
4. Importance weighting (optional).

Given the reference x and candidate $\hat{x}$, compute BERT embeddings and pairwise cosine similarity after highlighting the greedy matching and include the optional IDF importance weighting.

Recall with idf weighting is computed as follows in Equation (4):

$$R_{BERT} = \frac{\sum_{x_i \epsilon x} idf(x_i) max_{\hat{x}_j \epsilon \hat{x}} x_i^T \hat{x}_j}{\sum_{x_i \epsilon x} idf(x_i)}. \tag{4}$$

Since they used pre-normalized vectors, calculated scores have an identical numerical range of cosine similarity (between -1 and 1). However, in practice, observed scores are in a more limited range because of the learned geometry of contextual embeddings. To address this problem, authors propose rescaling using empirical lower bound b as a baseline, computed using Common Crawl monolingual datasets. Rescaled value can be computed as follows in Equation (5):

$$\hat{R}_{BERT} = \frac{R_{BERT} - b}{1 - b}. \tag{5}$$

After this operation $\hat{R}_{BERT}$ is typically between 0 and 1. This method does not affect the ranking ability and human correlation of BERTSCORE, and is intended solely to increase the score readability.

5.4 Recall-Oriented Understudy for Gisting Evaluation (ROUGE)

One of the most efficient summary evaluation metrics is ROUGE [63]. The metric was first proposed at the DUC conference in 2004. The main idea behind the ROUGE score is based on counting the number of n-gram matches between the candidate and the reference summaries. Rouge is one of the most popular metrics and is considered a generally accepted standard in summarization tasks.

There are variations for this metric in the scientific literature. The most common are ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S, a part of the publicly available NLTK natural language processing package for Python programming language [11].

Formally, ROUGE-N is an n-gram recall between a candidate and reference summaries, and it is computed as follows in Equation (6):

$$ROUGE{-}N = \frac{\sum_{S\epsilon\{RS\}} \sum_{gram_n \epsilon S} count_{match}\left(gram_n\right)}{\sum_{S\epsilon\{RS\}} \sum_{gram_n \epsilon S} count\left(gram_n\right)}, \tag{6}$$

where RS means Reference Summaries, n means the length of the n-gram, $gram_n$ and $Count_{match}(gram_n)$ is the maximum number of n-grams occurring both in a candidate and a reference summary.

ROUGE-L is the word Longest Common Subsequence (LCS) measure. The LCS advantage is that it does not require consecutive but in-sequence matches reflecting sentence-level word order. In addition, there is no need for a predefined n-gram length since LCS automatically includes the longest in-sequence common n-grams.

Rouge-l is based on the LCS score, which calculates the similarity between two abstracts, assuming X is a golden summary and Y is a candidate summary. ROUGE-L is computed as follows in Equation (9):

$$R_{lcs} = \frac{\mathsf{LCS}\left(X,Y\right)}{m}, \tag{7}$$

$$P_{lcs} = \frac{\mathsf{LCS}\left(X,Y\right)}{n}, \tag{8}$$

$$F_{lcs} = \frac{\left(1+\beta^2\right) R_{\mathsf{lcs}} P_{\mathsf{lcs}}}{R_{lcs} + \beta^2 P_{\mathsf{lcs}}}, \tag{9}$$

where LCS(X, Y) is the length measure of an LCS of X and Y, and β = $P_{\mathsf{lcs}}/R_{\mathsf{lcs}}$.

Unfortunately, the basic LCS needs to differentiate LCSes of different gaps between words in LCS within their embedding sequences. Thus, to improve the basic LCS, a new WLCS algorithm was introduced, which remembers the length of consecutive matches encountered so far to a regular two-dimensional dynamic program table computing LCS as in Equation (12):

$$R_{wlcs} = f^{-1}\frac{\mathsf{WLCS}\left(X,Y\right)}{f\left(m\right)}, \tag{10}$$

$$P_{wlcs} = f^{-1}\frac{\mathsf{WLCS}\left(X,Y\right)}{f\left(n\right)}, \tag{11}$$

$$F_{wlcs} = \frac{\left(1+\beta^2\right) R_{\mathsf{wlcs}} P_{\mathsf{wlcs}}}{R_{\mathsf{wlcs}} + \beta^2 P_{\mathsf{wlcs}}}, \tag{12}$$

where f^{-1} is the inverse function of weighting function f, with f it is possible to parameterize the WLCS algorithm to assign different weights to consecutive in-sequence matches, such that consecutive matches are given more scores than non-consecutive matches.

ROUGE-S, also known as skip-gram co-occurrence, allows for gaps between word pairs. For instance, skip-bigram measures the overlap between two words that are a maximum of two gaps apart.

Given summaries of length (X) m and n (Y), assuming X is a reference and Y is a candidate summary, skip-bigram-based F-measure can be computed as follows in Equation (15):

$$R_{skip2} = \frac{\mathsf{SKIP2}\left(X,Y\right)}{C\left(m,2\right)}, \tag{13}$$

$$P_{skip2} = \frac{\mathsf{SKIP2}\left(X,Y\right)}{C\left(n,2\right)}, \tag{14}$$

$$F_{skip2} = \frac{\left(1+\beta^2\right) R_{\mathsf{skip2}} P_{\mathsf{skip2}}}{R_{\mathsf{skip2}} + \beta^2 P_{\mathsf{skip2}}}, \tag{15}$$

where SKIP2(X,Y) is the skip-bigram matches number between X and Y, β controlling the relative importance of P_{skip2} and R_{skip2}, and C is the combination function.

[63] reported on the stability and reliability of ROUGE over various sample sizes. Nevertheless,

achieving a high correlation with human judgment in summarizing several documents is still an open research topic, as ROUGE has already done in summarizing a single document.

5.5 Pyramid

Pyramid is another choice to calibrate the metrics to human summarization manners [77].

The pyramid approach consists of two tasks:

1. Human annotators identify Summary Content Units (SCUs), the sets of text fragments expressing the same essential content, in model summaries and create a pyramid (SCUs are weighted according to the number of models in which they appear).
2. Evaluate a new summary against the Pyramid. The pyramid score is computed by the total weight of all SCUs present in the candidate divided by the total SCU weight possible for an average-length summary.

The Pyramid is a reliable and predictable metric. It helps to determine missing important parts and compare scores for different input sets. However, it has two main drawbacks:

1. The Pyramid metric ignores interdependencies between content units.
2. Creating an initial pyramid requires lots of work, and a large-scale application of the method would require an approach with a sufficient level of automation [77].

The Pyramid semi-automated algorithm consists of five steps:

1. Create initial Pyramid.
2. Enumerate all the candidate contributors(contiguous phrases) in each peer summary sentence.
3. Find the most similar SCU for each candidate.
4. Find a disjoint set of contributors with maximum total similarity with the Pyramid.
5. Calculate the pyramid summary score, employing the selected contributors and their SCU weights.

Suppose that the Pyramid has n tiers,T_n on top and T_1 on the bottom. The weights of SCUs in tier T_i will be i. Then, let $|T_i|$ denote the number of SCUs in tier $|T_i|$, and D_i be the number of SCUs, in summary, appearing in T_i. Summary SCUs not appearing in the Pyramid are assigned a zero weight. The total SCUs weight D is in Equation (16):

$$D = \sum_{i=1}^{n} i \times D_i. \tag{16}$$

The summary with X SCUs optimal content score is given in Equation (17):

$$Max = \sum_{i=j+1}^{n} i \times |T_i| + j \times \left(X - \sum_{i=j+1}^{n} |T_i| \right), \tag{17}$$

where j is given by $j = \max \left(\sum_{t=i}^{n} |T_t| \geq X \right)$.

5.6 Summarization Evaluation by Relevance Analysis (SERA)

The automated metric SERA, at a high level of generalization, evaluates the relevance score between the generated summary and the golden standard resume; the score is based on information retrieval approaches. As input, the algorithm can use keywords and phrases consisting of nouns, which can be obtained from the text of the generated resume. Keywords and phrases from nouns form queries for full-text search in the database of the gold standards summary. As a result of the search, the first few documents from the output ranked by relevance can be used to encode and compute human estimates of the quality of a text summary. This approach allows for terms not lexically equivalent but semantically related.

For scientific text summarization, SERA authors consider the scientific articles as the context for the words the articles consist of. Thus, if two words appear in similar articles, they are semantically related. Likewise, they consider the two summaries similar if they refer to the same article set, even if they do not share much lexical

content. The developers use information retrieval to see if a summary relates to an article, treating the summaries as queries and the articles as text documents. Then they rank the articles based on their relativity to a given summary. Numerically close article rankings suggest that the summaries are semantically related for a given pair of candidate and reference summaries [18].

SERA is defined as follows (18):

$$SERA = \frac{1}{M}\sum_{i=1}^{M}\frac{|R_C \cap R_{G_i}|}{|R_C|}. \tag{18}$$

Based on the focus domain, initially build an index from a set of related article texts. For example, given a candidate summary C and a set of reference summaries G_i, query the search engine with the candidate and gold summaries texts and compare their ranked results. R_C is the ranked list of fetched documents for candidate summary C, and R_G is the gold summary results ranked list.

5.7 Graph Distance (GRAD)

The motivation for developing another metric is to eliminate the shortcomings of the previous approaches [27]. The idea of the GRAD metric exploits a semantic graph of the input text. The nodes of a semantic graph are terms or words used in a text, and the weight of edges between nodes corresponds to the semantic relationship of adjacent nodes of words. In such a manner, the tested hypothesis is that a fair summary should contain words with corresponding nodes having the maximal number of neighbors from the source text in the semantic graph. Moreover, vice versa, if there are many terms with distance from source text nodes in the summary text, this summary should be scored lower. A summary quality measure is the inverted sum of weights for every term in a text to its closest term in the summary text. At least two summaries are required for every source text to calculate the metric.

GRAD authors state that a good summary consists of the terms referring to the central vertices in the semantic graph, meaning the terms are connected to the maximum number of other terms in a source text. With regards to the GRAD metric, the summary score is estimated as a normalized inverted distance sum from every text term to its closest summary term S as is in Equation (19):

$$score(S) = \frac{1}{|S|\sum_{v_i}\min_{v_j \epsilon V \cap S} d(v_j, v_i)}, \tag{19}$$

where $d(v_j, v_i)$ is the shortest path between v_i and v_j. Normalization is performed by dividing the score by the number of summary terms. Normalization is required to prevent the metric from the preference for longer summaries.

Additional findings are that the GRAD metric cannot discern generated summary text from other human-created summaries. Nevertheless, it can assess the similarity between them. The researchers suggest investigating the various extra features to improve GRAD metric performance, such as inverse document frequency of terms or part-of-speech tags.

5.8 Question Answering Evaluation (QA)

Recently, the evaluation metric based on QA correlates well with human judgments regarding coverage and focus on information is gaining popularity [21, 28].

In an ideal QA-based evaluation framework, a set of QA pairs represents all of the reference summary's information. The candidate summary's recall of this information is measured by answering the questions against the candidate.

The questions should be accountable if the information required to answer them is currently in the candidate. QA evaluation approach fundamentally differs from text overlap methods because it explicitly estimates how much of the reference's information is retained in the candidate.

QA evaluation approach consists of the following steps:

1. **Answer selection.** The first step is to select a set of phrases representing answers to questions that will be formed later. Answers should be chosen in such a way that they generate questions covering as much of the information in summary as possible.

2. **Question Generation** At this stage, a learned model generated a question for selected answers from the first step.
3. **Question Answering** a set of QA pairs was generated based on the reference summary from the previous steps. Moreover, the QA model is used to answer the questions against the candidate summary.

QA models for this approach have to decide whether a question is answerable to reduce noise from spurious answers because it is almost always the case that the candidate summary will not contain some reference summary information.

5.9 Bacronymic Language Model Approach for Summary Quality Estimation (BLANC)

Metric BLANC was proposed as the ROUGE family summary quality estimators substitute by [106].

BLANC can be defined as a numerical measure of how much a summary helps an independent language model to perform the understanding task on a source document. Authors focus on the masked token task, where a model is challenged to reconstruct masked text spans. To predict masked text tokens, BLANC authors used the BERT language model.

There are two versions of the BLANC metric:

1. **BLANC-help** directly concatenates the summary text to each sentence.
2. **BLANC-tune** finetunes the language model and processes the entire document using the summary text.

BLANC-help could be defined as follows in Equation (20):

$$BLANC_{help} = A_s - A_f = \frac{S_{01} - S_{10}}{S_{\text{total}}}. \quad (20)$$

After iterating over all sentences in a text and all possible masking combinations, the algorithm ends up with four counts of successful and unsuccessful unmaskings S_{ij}, i = 0, 1;j= 0, 1. Here, the index i equals 0 (unsuccessful unmasking) or 1 (successful unmasking) - for the filler input. The index j is defined similarly for the summary input. The BLANC values can range from -1 to 1, but the typical values are between 0 and 0.3.

6 Summarization Models

Automatic Text Summarization reduces the text size while preserving the core information. Text summarization models are usually classified as extractive or abstractive, single-document or multi-document; see Table 16. It is important to note that abstractive summarization models can form informative, indicative abstracts and a mix - it all depends on the dataset on which the model was trained.

Table 16. Description of the proposed models

Method	Abstractive	Extractive	Single-document	Multi-document
Luhn		✓	✓	
TextRank		✓	✓	
LexRank		✓	✓	
SumBasic		✓		✓
LSA		✓		✓
KL-sum		✓		✓
PEGASUS	✓		✓	
BigBird PEGASUS	✓		✓	
T5	✓			✓
BART	✓		✓	
HatBART	✓		✓	
GPT-2	✓		✓	
GPT-3	✓		✓	
SimCLS	✓		✓	
UniLM	✓		✓	

6.1 Conventional Extractive Text Summarization Models

These methods belong to the earliest attempts to automatically abstract texts in which the abstract consists entirely of sentences contained in the original text. However, there are also methods that use the procedure of smoothing of an extractive abstract, allowing to obtain a coherent text from disparate elements [9].

6.1.1 Luhn

One of the earliest examples of this type of summarization model was presented back in 1958 in a paper by [69]. Luhn Summarization algorithm's approach was based on scoring the text terms by frequency, selecting the sentences with the most important terms to construct a summary:

1. Ignore Stopwords: High-frequency words such as articles, prepositions, and pronouns, which do not carry the semantics but instead perform service functions in text, are ignored.
2. Determine Top Words: The document's high-frequency words are counted up.
3. Select Top Words: A relatively small number of the top-frequency words are selected for scoring.
4. Select Top Sentences: Scoring the sentences according to their top word content. The top four sentences are selected for the summary.

It is useful when very low-frequency and high-frequency words (stopwords) are insignificant.

This method can be considered the first discovery in text generalization. For example, the article on the Greedy Optimization Method to summarize scientific articles uses the basic idea of Luhn's extractive approach [2].

6.1.2 TextRank

[72] proposed a graph theory-based text summarization algorithm named TextRank [72], exploiting the concept of the previously known PageRank algorithm by Google [79], which represents the sentences in a text in the form of graph vertices and relations between sentences as edges. Each of the graph vertices is valued according to the semantic relatedness of the sentence to all other sentences in the text (similar to the number of hyperlinks to a page from other pages in the PageRank algorithm), calculated recursively from the entire graph Equation (21):

$$S(V_i) = (1-d) + d \times \sum_{j \epsilon In(V_i)} \frac{1}{|Out(V_j)|} S(V_j), \quad (21)$$

where V_i is a vertice, $In(V_i)$ is the set of vertices that point to it, and $Out(V_i)$ is the set of vertices that vertex points to, d is a damping factor that accepts values of 0 and 1, which carries the integrating role to calculate the probability of jumping from a given vertex to any other random vertex in the graph.

After calculating the similarity between all the sentences, a graph is built where each vertex might not be linked to any other vertex because of the lack of similarity between sentences represented by the vertices. The edges that connect two vertices will have a weight representing the force of the similarity. Finally, the algorithm summarizes the document using the most significant sentences and key phrases.

Authors of [52] enriched the algorithm of TextRank in terms of optimization problems by substituting conventional word encoding algorithms with word embeddings. In other words, the calculation of sentence similarity and the following graph is based on cosine similarity between word2vec [73] vectors of input sentences. The proposed approach is based on the Bayesian optimization of the function estimation based on the ROUGE estimation. The authors experimentally show that Bayesian finetuning of Textrank hyperparameters can outperform traditional models on the ROUGE-1, ROUGE-2, and ROUGE-L metrics. Experimental analysis shows that, with proper hyperparameter tuning, even an algorithm as simple as word2vec can significantly increase conventional algorithms' efficiency in the document summarization problem.

6.1.3 LexRank

LexRank algorithm was developed around 2004 by [26] at the University of Michigan. The algorithm scores the sentences by importance using the eigenvector centrality concept in the graph representation of sentences. In addition, it uses intra-sentence cosine similarity for the adjacency matrix of sentences.

The algorithm is based on graph theory. Sentences with removed stop-words in the text become the vertices of the graph, and edges are

constructed comparing sentence similarity using an IDF-modified-cosine in Equation (22):

$$idf\text{-}modified\text{-}cosine\,(x,y) = \frac{\sum_{w \epsilon x,y} tf_{w,x} tf_{w,y} \left(idf_w\right)^2}{\sqrt{\sum_{x_i \epsilon x} \left(tf_{x_i,x} idf_{x_i}\right)^2}} \times \frac{1}{\sqrt{\sum_{x_i \epsilon x} \left(tf_{y_i,y} idf_{y_i}\right)^2}}, \tag{22}$$

where $tf_{w,s}$ is the number of occurrences of the word w in the sentence s and $idf_w = \log\left(\frac{N}{n_w}\right)$.

After the graph is constructed, Google's PageRank [79] algorithm is applied. So, highly ranked sentences are similar to many other sentences in the text. The resulting summary is created by choosing the highest-rated x sentences where the user defines x as the wanted number of sentences in the summary.

As a further development of the method, an approach combining machine learning, text graph representation, and linguistic knowledge base was developed in 2020 [7].

6.1.4 SumBasic

SumBasic is an algorithm that selects sentences based on the frequency with a re-weight component for the word probabilities to minimize redundancy [78].

In SumBasic, each sentence S is assigned a score based on its high-frequency word content Equation (23):

$$Score\,(S) = \sum_{w \epsilon S} \frac{1}{|S|} P_D\,(w)\,, \tag{23}$$

where P_D is the observed unigram probabilities obtained from the document collection D. A summary is progressively built by adding the highest-scoring sentence. To avoid redundancy, the importance of the word in the selected sentence is updated $P^D_{new}\,(w) = P^D_{old}\,(w)^2$. Sentences are selected in this manner until we reach the summary word limit.

6.1.5 Latent Semantic Analysis (LSA)

LSA is a mathematic-statistical method that extracts hidden semantic structures of words and sentences in an unsupervised way [50]. LSA uses the input document context and captures word co-occurrence and what common words are used in various sentences.

A significant number of words co-occurring among sentences indicates that they are semantically related. It is because the meaning of a sentence is determined by the words it contains, and the meanings of words are defined by the other words within a sentence, defining the context.

Singular Value Decomposition (SVD), an algebraic method, is used to find the interrelations between sentences and words [35]. Besides modeling relationships between words and sentences, SVD can also reduce noise to improve accuracy.

The paper [42] demonstrates the union of conventional and modern NLP algorithms, namely LSA [50], BERT [22]. The algorithm proposed in this paper can obtain a higher score than summarizing the text using topic modeling with a Latent Dirichlet distribution (LDA). An experiment in which the proposed research will summarize a long text document using LSA topic modeling along with a TFIDF keyword extractor for each sentence in the text document and using the BERT encoder model to encode sentences from the text document in order. Another approach developed in 2020, additionally to the topic modeling, uses machine learning and rhetoric analysis exploiting discourse markers [8].

6.1.6 Kullback-Leibler (KL) Sum Algorithm

KL Sum algorithm selects sentences from the source text with a summary length fixed to L. Then, it adds sentences to a summary greedily as long as it decreases the KL Divergence. The objective of the KL Sum algorithm is to find sentences set whose length is less than L words and whose unigram distribution is close to that of the source document [45].

In mathematical statistics, the KL divergence (or relative entropy) measures how two probability distributions differ. The smaller the divergence, the

more similar the summary is to the document by readability and the meaning carried [57].

The KL introduces a summary selection criterion for sentences to include in S given sentence collection in a document D as shown in Equation (24):

$$S^* = min(S : words(S) \leq L, KL(P_D||P_S)), \quad (24)$$

where P_S is the empirical unigram distribution of the candidate summary S and $KL\,(P||Q)$ represents the Kullback-Leibler (KL) divergence given by $\sum_w P\,(w) \log \frac{P(w)}{Q(w)}$. This value represents the divergence between the true distribution P and the approximated distribution Q.

This criterion treats the text summarization as finding a set of summary sentences from the source text that closely match the source unigram distribution.

6.2 Modern Extractive Text Summarization Models

6.2.1 Graph Based Approach

The researchers have used graph-based ideas since the conventional era of extractive summarization. Nowadays, plenty of works with graph approach [24,82,83,111]. The central concept is to present the text in a graph highlighting its semantic and syntax structures. The graph structure gives valuable information that can solve open problems like improvement of text cohesion and factuality. An additional boost in solving the problems is using modern language models.

Paper [111] addresses the problem of summarization factuality by extracting fact-level semantic units to improve the overall performance of the summarization model. The authors incorporate their model with word embedding called BERT [22] using a hierarchical graph mask that leads to combining embeddings' ability in natural language understanding and the structural information without increasing the scale of the model. There are a couple of interesting findings. Namely, the first one is that combining sentence-level relationships, semantic units, and document-level information leads to poor results. The authors suggested the potential cause of such phenomena is that the document-level information needs to be more effective for single document summarization. Another author finding is that the golden summaries are distributed smoothly across documents on the CNN/DailyMail dataset [74]. In contrast, the summaries generated by models are highly biased towards the beginning of texts.

6.2.2 Deep Learning Based Approach

Technological convergence suggests the interpenetration of ideas between different areas of science and technology, e.g., the approach of artificial neural networks found its advantage in open problems of extractive summarizations compared with the conventional approach [66, 99, 102, 110]. Since an artificial neural network is a universal approximator, the idea of implementing it for summarization is pretty obvious.

Moreover, the simplicity of the research and development cycle in the framework of artificial neural networks can lead to the result [102]. Authors stated that their approach, with the help of Long Short Term Memory (LSTM) [48] and simple word embeddings overcome the existing models in the same class of algorithms, namely their model achieved an average F1-Score of 0.84the nearest conventional extractive method LexRank with average F1-Score 0.80.

In the paper, [66] authors demonstrated the application of BERT [22] in both text summarization tasks, namely extractive and abstractive. The proposed extractive model is based on the BERT document-level encoder that maps sentences to their vector representations while preserving semantic features. The authors applied the conventional technique of stacking layers to improve the model's performance.

One of the authors' findings is that a two-staged finetuning approach can further boost the quality of the generated summaries. Due to the pre-trained feature of BERT, the suggested model achieved state-of-the-art results across the board extractive settings.

6.3 Abstractive Text Summarization Models

6.3.1 Pre-training with Extracted Gap-sentences for Abstractive Summarization Sequence-to-sequence (PEGASUS)

PEGASUS is based on the seq2seq architecture like any other sequence transformer task. Nevertheless, the novelty of this architecture lies in its use of a self-supervised objective to train a transformer model called Gap Sentences Generation (GSG) [114].

The model masks meaningful sentences from an input document and generates them as an output sequence with all the remaining sentences. The PEGASUS masks those sentences from the text most similar to the reference summary sentences. So, predicting such sentences would maximize the ROUGE score of the candidate summary.

Although the PEGASUS's main contribution is the GSG, it has a transformer architecture; thus, it makes sense to pre-train the encoder as a Masked Language Model (MLM). The MLM randomly masks sequence words and uses other sequences to predict those masked words. The GSG task can be seen as a document-level MLM derived from this concept.

6.3.2 BigBird PEGASUS

The BigBird is a sparse-attention-based transformer extending transformer-based models, such as BERT, to a much longer sequence. Besides sparse attention, BigBird also has global attention and random attention to the input sequence [112].

Theoretically, it has been shown that applying sparse, global, and random attention approximately performs on the level of full attention while being computationally less complex for longer text sequences. Furthermore, BigBird showed improved summarization performance compared to BERT or RoBERTa due to the ability to handle a broader context.

6.3.3 Text-to-Text-Transfer-Transformer (T5)

T5 model suggests dealing with all the NLP tasks in a unified text-to-text format when both the input and output are text strings [88]. T5 model is an equivalent of the original Transformer proposed by [107].

The subtle difference the T5 model employs from previously trained MLM models is in replacing several consecutive tokens with a single Mask keyword. During T5 pre-training, it transforms the original text into Input and Output pairs by adding noise.

6.3.4 Bidirectional and Auto-Regressive Transformer (BART)

Recently introduced BART consists of two major components: a bidirectional encoder and an autoregressive decoder, which have a transformer-based architecture and are implemented as a seq2seq model [60].

The Basic BART model uses six layers in both the encoder and the decoder, while the Large model has 12 layers. When pre-training BART mode, the following techniques are applied:

1. Token Masking: a random subset of the input is replaced with [MASK] tokens, just like in the BERT model.
2. Token Deletion: random tokens are deleted from the input, and the model must decide what is missing.
3. Text Infilling: a number of varying length text spans are replaced with a single [MASK] token.
4. Sentence Permutation: shuffling of the input sentences.
5. Document Rotation: a token is chosen at random, and the sequence is rotated to start with the token chosen.

6.3.5 Hierarchical Attention BART (HatBART)

HatBART is a new hierarchical attention Transformer-based architecture that outperforms standard Transformers on several seq2seq tasks [91].

Authors modified the standard sequence to sequence transformer architecture [107] by adding hierarchical attention for improved processing of long documents. The number of parameters for a large hierarchical model on summary tasks is 471M compared to the plain transformer 408M.

The twelve encoder and decoder layers were used, a hidden size of 1024 4096 for the dimension of the fully connected feed-forward networks and 16 attention heads both in the encoder and the decoder. Unlike the original Transformer, GELU activation is used instead of ReLU.

6.3.6 Generative Pre-trained Transformer (GPT)

GPT-2 GPT-2 is a huge transformer-based language model having 1.5B parameters, trained on an 8M web pages dataset [4]. GPT-2 is trained with the objective of predicting the next word, taking into consideration the previous words in a text. It uses Byte Pair Encoding (BPE) for its token vocabulary construction, which means they are usually word parts and output one token at a time.

The model receives only one input token, so only one path would be active. The token is processed consequently through all the layers, and then a vector is an output, which can be scored using the model's vocabulary. In this case, the token with the highest probability is selected. In addition, GPT-2 has a parameter called top-k that can be used to have the model consider sampling words other than the top word. Next, add the first step output to the input sequence and have the model make the following prediction.

Each GPT-2 layer retains the first token interpretation and uses it in processing the second token. Thus, GPT-2 does not re-interpret the first token in light of the second token.

GPT-3 At its core, GPT-3 is a transformer model, a seq2seq deep learning model producing a text sequence given an input sequence. The models of this type are designed for text generation tasks such as Question Answering (QA), Text Summarization (TS), and Machine Translation (MT). GPT-3 is the third-generation GPT language model by OpenAI. The main difference that sets GPT-3 from previous models is its enormous size. GPT-3 contains 175B parameters, making it 17 times larger than its GPT-2 predecessor and about ten times larger than Microsoft's Turing NLG model [13].

GPT-3 capacity is by three orders of magnitude than GPT-2 with no significant change in model architecture, just more numerous and broader layers with more training data.

6.3.7 Simple Framework for Contrastive Learning of Abstractive Summarization (SimCLS)

SimCLS is a conceptually simple framework for abstractive text summarization. The model bridges the gap between the learning objective and evaluation metrics, resulting from the currently dominating seq2seq learning framework by treating text generation as a reference-free evaluation problem assisted by contrastive learning [67].

SimCLS framework for two-stage abstractive summarization:

1. BART is used for candidate summary generation.
2. The RoBERTa scoring model is used to predict the quality of the candidate summaries based on the content of the source document.

6.3.8 Unified Pre-trained Language Model (UniLM)

UniLM is a multi-layer NN made up of several Transformer AI models jointly pre-trained on large text data amounts and optimized for language modeling. Models have attention in a way that each output element is connected to every input element, and as a result, the weightings between them are calculated dynamically.

The pre-trained UniLM is similar to BERT, and on-demand, it can be fine-tuned to adapt to various downstream NLP tasks. UniLM can be configured in distinction from BERT using different self-attention masks to aggregate the context for different language models [23]. Additionally, due to their unified nature, the Transformer networks can share parameters, which makes learned text representations more general and thus mitigates overfitting to any single task.

7 Results

All these models demonstrated significant results in text summarization. In Table 17 and Table 18 we present the evaluation of the six extractive algorithms described over DUC2001, CNN/Daily Mail, XSum, and BigPatent datasets based on ROUGE-1 and ROUGE-2[11] metrics.

According to the ROUGE-1 and ROUGE-2 metrics on the DUC2001, CNN/Daily Mail, XSum, and BigPatent datasets three most successful models can be distinguished: Luhn, TexRank, and LexRank.

Interestingly, on the XSum dataset, the SumBasic model has a maximum ROUGE-1 indicator of only 18.56, and a LexRank model's maximum value of ROUGE-2 is equal to 3.

Table 17. Extractive summarization models result on DUC2001, CNN/Daily Mail, XSum datasets. R1 and R2 stand for the ROUGE-1 and ROUGE-2 respectively

	DUC2001		CNN/Daily Mail		XSum	
	R-1	R-2	R-1	R-2	R-1	R-2
Luhn	42.07	**16.81**	/	/	/	/
TextRank	40.42	15.40	**40.20**	**17.56**	/	/
LexRank	**42.30**	15.80	35.34	13.31	17.95	3.00
LSA	35.85	11.98	/	/	/	/
SumBasic	36.03	11.24	34.11	11.13	**18.56**	2.91
KLSum	35.85	11.70	29.92	10.50	16.73	2.83

Evaluation of the popular abstractive algorithms is presented in Table 19 and Table 20. In addition, the quality assessment of the summarization models is carried out on the largest datasets.

[11] We have omitted ROUGE-L metric as it strongly correlates with the ROUGE-1

Table 18. Extractive summarization models'result on BigPatent, ArXiv,PubMed datasets. R1 and R2 stand for the ROUGE-1 and ROUGE-2 respectively

	BigPatent		ArXiv		PubMed	
	R-1	R-2	R-1	R-2	R-1	R-2
Luhn	/	/	/	/	/	/
TextRank	**35.99**	**11.14**	/	/	/	/
LexRank	35.57	10.47	**33.85**	**10.73**	**39.19**	**13.89**
LSA	/	/	29.91	7.42	33.89	9.93
SumBasic	27.44	7.08	29.47	6.95	37.15	11.36
KLSum	/	/	/	/	/	/

It can be noted that the SimCLS algorithm with the R-1 score of 46.67 and 47.61 is the leader among other models, including extractive algorithms. By the ROUGE-1 and ROUGE-2 metrics, the BigBird PEGASUS and PEGASUS models are also in the top three.

The other algorithms also show good summarization ability, lagging behind the leaders by 1-2 points in the R-1 and R-2 metrics.

After reviewing the state-of-the-art text summarization models, we found that, in general, abstractive models outperform extractive ones based on the ROUGE-1 metric; see Fig. 6.

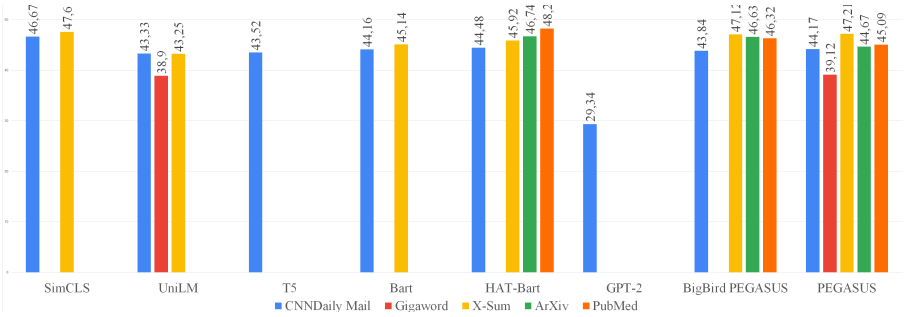
8 Conclusion

Text Summarization is an exciting research topic in the NLP community that helps humans process large amounts of information producing meaningful extracts. This article aims to present the latest research and advances in this area.

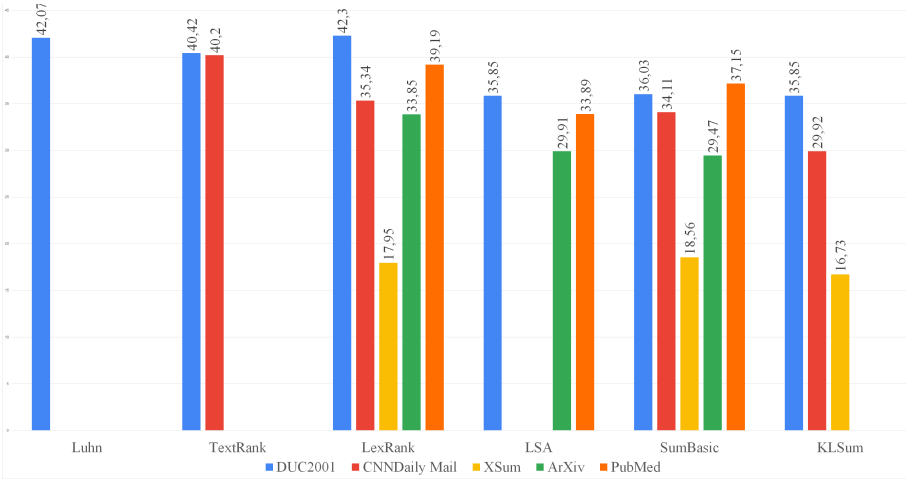
The review's important thing that is considered attractive is the analysis results, which state that extractive summaries are relatively more straightforward than abstractive summaries, which are very complex. Moreover, the latest research shows that abstractive models have higher ROUGE-1 and ROUGE-2 metrics than extractive models on the same dataset. By the way, for some tasks, such as question summarization, questions that extractive approaches cannot summarize, but abstractive are required [51]. However, extractive summaries are still the topic for research, as there are still many challenging things for researchers.

Table 19. Abstractive summarization models result on CNN/Daily Mail, Gigaword, X-Sumdatasets. R1 and R2 stand for ROUGE-1 and ROUGE-2, respectively.

	CNN/Daily Mail		Gigaword		X-Sum	
	R-1	R-2	R-1	R-2	R-1	R-2
SimCLS	**46.67**	**22.15**	/	/	**47.61**	**24.57**
UniLM	43.33	20.21	38.90	**20.05**	43.25	20.55
T5	43.52	21.55	/	/	/	/
Bart	44.16	21.28	/	/	45.14	22.27
HAT-Bart	44.48	21.31	/	/	45.92	22.79
GPT-2	29.34	8.27	/	/	/	/
BigBird PEGASUS	43.84	21.11	/	/	47.12	24.05
PEGASUS	44.17	21.47	**39.12**	19.86	47.21	24.56



(a) Abstractive summarization models



(b) Extractive summarization models

Fig. 6. ROUGE-1 for Abstractive (a) and Extractive (b) summarization models on CNN/Daily Mail, Gigaword, X-Sum, BigPatent, ArXiv, and PubMed datasets

We believe abstractive summarization models are superior to extractive ones because they are trained on datasets where the summaries are abstractive. Models mock them and express the summary similarly to the reference summary, thus giving a high ROUGE scoring [66].

Table 20. Abstractive summarization models result on ArXiv and PubMed datasets. R1 and R2 stand for ROUGE-1 and ROUGE-2, respectively

	ArXiv		PubMed	
	R-1	R-2	R-1	R-2
SimCLS	/	/	/	/
UniLM	/	/	/	/
T5	/	/	/	/
Bart	/	/	/	/
HAT-Bart	**46.74**	**19.19**	**48.25**	**21.35**
GPT-2	/	/	/	/
BigBird PEGASUS	46.63	19.02	46.32	20.65
PEGASUS	44.67	/	45.09	/

Finally, we need a new metric to assess summarization quality: a metric that counts the number of n-grams generated and the reference summaries shared and measures the semantic similarity between the summary texts. It should also show how much information from the original text is retained and the compression level.

As for the study of text summarization, future work can be performed on the following tasks:

1. Selection of standards for the formation of Golden Summaries.
2. Creating an objective metric for the summarization of the text.
3. Challenge of increasing the maximum ROUGE-1 and ROUGE-2 values.

9 List of Abbreviations

In this section, we present the list of abbreviations used in the paper:

— ABS–attention-based summarization

— AI–Artificial Intelligence

— B–billion (1 000 000 000)

— BART–Bidirectional and Auto-Regressive Transformer

— BERT –Bidirectional Encoder Representations from Transformers

— BLANC–Bacronymic Language model Approach for summary quality estimation. Cool?

— BLEU –Bilingual Evaluation Understudy

— BP–Brevity Penalty

— BPE–Byte Pair Encoding

— CLTS–Cross-Language Text Summarization

— DUC–Document Understanding Conference

— FRUMP–Fast Reading Understanding and Memory Program

— GiB–Gibibyte

— GPO–US Government Publishing Office

— GPT–Generative Pre-trained Transformer

— GRAD–GRAph Distance

— GSG–Gap Sentences Generation

— h-index–Hirsch-Index

— HatBART–Hierarchical attention BART

— HMM–hidden Markov models

— K–thousands (from the kilo in Greek)

— KiB–Kibibyte

— KL–Kullback-Leibler

— LCS–Longest Common Subsequence

— LNAI–Lecture Notes in Artificial Intelligence

— LNBI–Lecture Notes in Bioinformatics

— LNCS–Lecture Notes in Computer Science

— LSA–Latent Semantic Analysis

— M–million

— MLM–Masked Language Model

— MT–Machine Translation

— NLG–Natural Language Generation

— NLM–National Library of Medicine's

— NLTK–Natural Language Tool Kit

— QA–Question Answering

— RANLP–International Conference on Recent Advances In Natural Language Processing

— RoBERTa–Robustly Optimized BERT Pre-training Approach

— ROUGE –Recall-Oriented Understudy for Gisting Evaluation

— SE–Search Engine

— SCUs–Summary Content Units

— Seq2Seq –Sequence to sequence neural network architecture

— SERA–Summarization Evaluation by Relevance Analysis

— SimCLS–Simple Framework for Contrastive Learning of Abstractive Summarization

— SVD–Singular Value Decomposition

— T5–Text-to-Text-Transfer-Transformer

— TAC–Text Analysis Conference

— TC–Total Citations

— TF-IDF–Term Frequency-Inverse Document Frequency

— THE–Times Higher Education

— TP–Total Publications

— TS–Text Summarization

— TSC–Citations number of publications in the field of Text Summarization

— TSP–Number of articles on Text Summarization

— UniLM–Unified pre-trained Language Model

— X-Sum–Extreme Summarization

Acknowledgements

This work was supported by the Ministry of Education and Sciences of the Republic of Kazakhstan under the following grants: #AP09260670 “Development of methods and algorithms for augmenting input data for modifying vector word embeddings”, and #AP14871214, “Development of machine learning methods, to increase the coherence of text in summaries produced by the Extractive Summarization Methods”. The funders had no role in study design, data collection and analysis, publication decisions, or manuscript preparation.

References

1. **Abualigah, L., Bashabsheh, M. Q., Alabool, H., Shehab, M. (2020).** Text summarization: A brief review. Studies in Computational Intelligence, Vol. 874, No. January, pp. 1–15.

2. **Akhmetov, I., Gelbukh, A., Mussabayev, R. (2021).** Greedy optimization method for extractive summarization of scientific articles. IEEE Access, pp. 1–1. DOI: 10.1109/ACCESS.2021.3136302.

3. **Akhmetov, I., Mladenovic, N., Mussabayev, R. (2021).** Using k-means and variable neighborhood search for automatic summarization of scientific articles. **Mladenovic, N., Sleptchenko, A., Sifaleras, A., Omar, M.**, editors, Variable Neighborhood Search, Springer International Publishing, Cham, pp. 166–175.

4. **Alec, R., Jeffrey, W., Rewon, C., David, L., Dario, A., Ilya, S. (2019).** Language models are unsupervised multitask learners, enhanced reader. OpenAI Blog, Vol. 1.

5. **Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., Kochut, K. (2017).** Text summarization techniques: A brief survey.

6. **Barzilay, R., McKeown, K. R. (2005).** Sentence fusion for multidocument news summarization. Comput. Linguist., Vol. 31, No. 3, pp. 297–328.
7. **Batura, T., Bakiyeva, A. (2020).** Hybrid approach to automatic summarization of scientific and technical texts. Journal of Theoretical and Applied Information Technology, pp. 559–570.
8. **Batura, T., Bakiyeva, A., Charintseva, M. (2020).** A method for automatic text summarization based on rhetorical analysis and topic modeling. International Journal of Computing, pp. 118–127. DOI: 10.47839/ijc.19.1.1700.
9. **Batura, T. V., , Bakiyeva, A. M., and (2018).** Developing the system for automatic summarization of scientific texts. Vestnik NSU. Series: Information Technologies, Vol. 16, No. 3, pp. 74–86. DOI: 10.25205/1818-7900-2018-16-3-74-86.
10. **Bhatia, N., Jaiswal, A. (2016).** Automatic text summarization and it's methods - A review. Proceedings of the 2016 6th International Conference - Cloud System and Big Data Engineering, Confluence 2016, pp. 65–72. DOI: 10.1109/CONFLUENCE.2016.7508049.
11. **Bird, S., Klein, E., Loper, E. (2009).** Natural language processing with Python: analyzing text with the natural language toolkit. O'Reilly Media, Inc.
12. **Bojanowski, P., Grave, E., Joulin, A., Mikolov, T. (2017).** Enriching word vectors with subword information. Transactions of the association for computational linguistics, Vol. 5, pp. 135–146.
13. **Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D. (2020).** Language models are few-shot learners.
14. **Carbonell, J., Stewart, J. (1999).** The use of MMR, diversity-based reranking for reordering documents and producing summaries. SIGIR Forum (ACM Special Interest Group on Information Retrieval). DOI: 10.1145/290941.291025.
15. **Celikyilmaz, A., Bosselut, A., He, X., Choi, Y. (2018).** Deep communicating agents for abstractive summarization.
16. **Chen, Y., Ma, Y., Mao, X., Li, Q. (2019).** Multi-task learning for abstractive and extractive summarization. Data Science and Engineering. DOI: 10.1007/s41019-019-0087-7.
17. **Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., Goharian, N. (2018).** A discourse-aware attention model for abstractive summarization of long documents. NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, pp. 615–621. DOI: 10.18653/v1/n18-2097.
18. **Cohan, A., Goharian, N. (2016).** Revisiting summarization evaluation for scientific articles.
19. **Conroy, J. M., O'Leary, D. P. (2001).** Text summarization via hidden Markov models. SIGIR '01, pp. 406–407.
20. **DeJong, G. (1979).** Prediction and substantiation: A new approach to natural language processing. Cognitive Science. DOI: 10.1016/S0364-0213(79)80009-9.
21. **Deutsch, D., Bedrax-Weiss, T., Roth, D. (2020).** Towards question-answering as an automatic metric for evaluating the content quality of a summary. CoRR, Vol. abs/2010.00490.
22. **Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. (2019).** BERT: Pre-training of deep bidirectional transformers for

language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

23. **Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., Hon, H.-W. (2019).** Unified language model pre-training for natural language understanding and generation.
24. **Dutta, M., Das, A. K., Mallick, C., Sarkar, A., Das, A. K. (2019).** A graph based approach on extractive summarization. In Emerging Technologies in Data Mining and Information Security. Springer, pp. 179–187.
25. **El-Refaiy, A., Abas, A., Elhenawy, I. (2018).** Review of recent techniques for extractive text summarization. Journal of Theoretical and Applied Information Technology, Vol. 96, pp. 7739–7759.
26. **Erkan, G., Radev, D. R. (2004).** LexRank: Graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research, Vol. 22, pp. 457–479. DOI: 10.1613/jair.1523.
27. **Ermakova, L., Firsov, A. (2018).** GRAD: A metric for evaluating summaries. **Mothe, J., Cellier, P., Ligozat, A.**, editors, COnférence en Recherche d'Informations et Applications - CORIA 2018, 15th French Information Retrieval Conference, Rennes, France, May 16-18, 2018. Proceedings, ARIA. DOI: 10.24348/coria.2018.paper6.
28. **Eyal, M., Baumel, T., Elhadad, M. (2019).** Question answering as an automatic evaluation metric for news article summarization. CoRR, Vol. abs/1906.00318.
29. **Ferreira, R., Cabral, L., Lins, R., Silva, G., Freitas, F., Cavalcanti, G., Lima, R., Simske, S., Favaro, L. (2013).** Assessing sentence scoring techniques for extractive text summarization. Expert Systems with Applications, Vol. 40, pp. 5755–5764. DOI: 10.1016/j.eswa.2013.04.023.
30. **Flores, M. L., Santos, E. R., Silveira, R. A. (2019).** Ontology-based extractive text summarization: The contribution of instances. Computacion y Sistemas, Vol. 23, pp. 905–914. DOI: 10.13053/cys-23-3-3270.
31. **Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., Presser, S., Leahy, C. (2020).** The Pile: An 800 GB dataset of diverse text for language modeling. arXiv preprint arXiv:2101.00027.
32. **Gehrmann, S., Deng, Y., Rush, A. M. (2020).** Bottom-up abstractive summarization. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018, pp. 4098–4109. DOI: 10.18653/v1/d18-1443.
33. **Gholamrezazadeh, S., Salehi, M. A., Gholamzadeh, B. (2009).** A comprehensive survey on text summarization systems. 2009 2nd International Conference on Computer Science and its Applications, IEEE, pp. 1–6.
34. **Giannakopoulos, G., Karkaletsis, V. (2013).** Summary evaluation: Together we stand npower-ed. International Conference on Intelligent Text Processing and Computational Linguistics, Springer, pp. 436–450.
35. **Gliozzo, A., Giuliano, C., Strapparava, C. (2005).** Domain kernels for word sense disambiguation. Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), Association for Computational Linguistics, Ann Arbor, Michigan, pp. 403–410. DOI: 10.3115/1219840.1219890.
36. **Goldstein, J., Kantrowitz, M., Mittal, V., Carbonell, J. (1999).** Summarizing text documents: Sentence selection and evaluation metrics. Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pp. 121–128.

37. **Gong, Y., Liu, X. (2001).** Generic text summarization using relevance measure and latent semantic analysis. Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Association for Computing Machinery, New York, NY, USA, pp. 19–25. DOI: 10.1145/383952.383955.

38. **Graff, D., Cieri, C. (2003).** English Gigaword.

39. **Greene, D., Cunningham, P. (2006).** Practical solutions to the problem of diagonal dominance in kernel document clustering. Proceedings of the 23rd International Conference on Machine Learning, Association for Computing Machinery, New York, NY, USA, pp. 377–384. DOI: 10.1145/1143844.1143892.

40. **Grusky, M., Naaman, M., Artzi, Y. (2018).** NEWSROOM: A dataset of 1.3 million summaries with diverse extractive strategies. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, New Orleans, Louisiana, pp. 708–719.

41. **Gu, J., Lu, Z., Li, H., Li, V. O. K. (2016).** Incorporating copying mechanism in sequence-to-sequence learning.

42. **Gupta, H., Patel, M. (2021).** Method of text summarization using LSA and sentence based topic modelling with Bert. 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), IEEE, pp. 511–517.

43. **Gupta, S., Gupta, S. (2018).** Abstractive summarization: An overview of the state of the art. Expert Systems with Applications, Vol. 121. DOI: 10.1016/j.eswa.2018.12.011.

44. **Gutierrez-Hinojosa, S. J., Calvo, H., Moreno-Armendariz, M. A. (2019).** Central embeddings for extractive summarization based on similarity. Computacion y Sistemas, Vol. 23, pp. 649–663. DOI: 10.13053/cys-23-3-3256.

45. **Haghighi, A., Vanderwende, L. (2009).** Exploring content models for multi-document summarization. Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Boulder, Colorado, pp. 362–370.

46. **Hearst, M. A. (1997).** TextTiling: Segmenting text into multi-paragraph subtopic passages. Comput. Linguist., Vol. 23, No. 1, pp. 33–64.

47. **Hinton, G. E., Osindero, S., Teh, Y.-W. (2006).** A fast learning algorithm for deep belief nets. Neural Computation, Vol. 18, No. 7, pp. 1527–1554. DOI: 10.1162/neco.2006.18.7.1527.

48. **Hochreiter, S., Schmidhuber, J. (1997).** Long short-term memory. Neural computation, Vol. 9, pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

49. **Hu, M., Liu, B. (2004).** Mining and summarizing customer reviews. Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, New York, NY, USA, pp. 168–177. DOI: 10.1145/1014052.1014073.

50. **Hu, X., Cai, Z., Louwerse, M. M., Olney, A. M., Penumatsa, P., Graesser, A. C. (2003).** A revised algorithm for latent semantic analysis. International Joint Conference on Artificial Intelligence, pp. 1489–1491.

51. **Ishigaki, T., Takamura, H., Okumura, M. (2019).** Extractive and abstractive summarization for multiple-sentence questions. Journal of Natural Language Processing, Vol. 26, pp. 37–58. DOI: 10.5715/jnlp.26.37.

52. **Jain, D., Borah, M. D., Biswas, A. (2020).** Fine-tuning textrank for legal document summarization: A Bayesian optimization based approach. Forum for Information Retrieval Evaluation, pp. 41–48.

53. **Khan, A., Salim, N., Jaya Kumar, Y. (2015).** A framework for multi-document abstractive summarization based on semantic role labelling. Applied Soft Computing, Vol. 30. DOI: 10.1016/j.asoc.2015.01.070.

54. **Khan, R., Qian, Y., Naeem, S. (2019).** Extractive based text summarization using KMeans and TF-IDF. International Journal of Information Engineering and Electronic Business, Vol. 11, pp. 33–44. DOI: 10.5815/ijieeb.2019.03.05.

55. **Kornilova, A., Eidelman, V. (2019).** Billsum: A corpus for automatic summarization of us legislation.

56. **Koupaee, M., Wang, W. Y. (2018).** WikiHow: A large scale text summarization dataset.

57. **Kullback, S., Leibler, R. A. (1951).** On information and sufficiency. The Annals of Mathematical Statistics, Vol. 22, No. 1, pp. 79–86.

58. **Kupiec, J., Pedersen, J., Chen, F. (1995).** Trainable document summarizer. SIGIR Forum (ACM Special Interest Group on Information Retrieval), pp. 68–73. DOI: 10.1145/215206.215333.

59. **Kupiec, J., Pedersen, J., Chen, F. (1995).** A trainable document summarizer. Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Association for Computing Machinery, New York, NY, USA, pp. 68–73. DOI: 10.1145/215206.215333.

60. **Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L. (2020).** BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, pp. 7871–7880. DOI: 10.18653/v1/2020.acl-main.703.

61. **Li, P., Lam, W., Bing, L., Wang, Z. (2017).** Deep recurrent generative decoder for abstractive text summarization.

62. **Li, W., McCallum, A. (2006).** Pachinko allocation: DAG-structured mixture models of topic correlations. Proceedings of the 23rd International Conference on Machine Learning, Association for Computing Machinery, New York, NY, USA, pp. 577–584. DOI: 10.1145/1143844.1143917.

63. **Lin, C.-Y. (2004).** Rouge: A package for automatic evaluation of summaries. Text summarization branches out, pp. 74–81.

64. **Lin, C.-Y., Hovy, E. (2003).** Automatic evaluation of summaries using n-gram co-occurrence statistics. Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, pp. 150–157.

65. **Liu, F., Flanigan, J., Thomson, S., Sadeh, N., Smith, N. A. (2018).** Toward abstractive summarization using semantic representations.

66. **Liu, Y., Lapata, M. (2020).** Text summarization with pretrained encoders. EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference, pp. 3730–3740. DOI: 10.18653/v1/d19-1387.

67. **Liu, Y., Liu, P. (2021).** SimCLS: A simple framework for contrastive learning of abstractive summarization.

68. **Lloret, E., Sanz, M. (2012).** Text summarisation in progress: A literature review. Artif. Intell. Rev., Vol. 37, pp. 1–41. DOI: 10.1007/s10462-011-9216-z.

69. **Luhn, H. P. (1958).** The automatic creation of literature abstracts. IBM Journal of Research and Development, Vol. 2, No. 2, pp. 159–165. DOI: 10.1147/rd.22.0159.

70. **Mahalakshmi, P., Fatima, N. S., Saravanan, V., Arshad, M. (2021).** Cross-language based multi-document summarization model using machine learning technique. Turkish Journal of Computer and Mathematics Education (TURCOMAT), Vol. 12, No. 6, pp. 331–335. DOI: 10.17762/turcomat.v12i6.1393.

71. **Mei, Q., Ling, X., Wondra, M., Su, H., Zhai, C. (2007).** Topic sentiment mixture: Modeling facets and opinions in weblogs. Proceedings of the 16th International Conference on World Wide Web, Association for Computing Machinery, New York, NY, USA, pp. 171–180. DOI: 10.1145/1242572.1242596.

72. **Mihalcea, R., Tarau, P. (2004).** TextRank: Bringing order into text. Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL, ACL, Barcelona, Spain, pp. 404–411.

73. **Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J. (2013).** Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, Vol. 26.

74. **Nallapati, R., Zhou, B., dos Santos, C., Gulçehre, Ç., Xiang, B. (2016).** Abstractive text summarization using sequence-to-sequence RNNs and beyond. CoNLL 2016 - 20th SIGNLL Conference on Computational Natural Language Learning, Proceedings, pp. 280–290. DOI: 10.18653/v1/k16-1028.

75. **Narayan, S., Cohen, S. B., Lapata, M. (2020).** Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018, pp. 1797–1807. DOI: 10.5281/zenodo.2399762.

76. **Nazari, N., Mahdavi, M. A. (2019).** A survey on automatic text summarization. Journal of AI and Data Mining, Vol. 7, pp. 121–135.

77. **Nenkova, A., Passonneau, R. J. (2004).** Evaluating content selection in summarization: The pyramid method. Proceedings of the human language technology conference of the north american chapter of the association for computational linguistics: Hlt-naacl 2004, pp. 145–152.

78. **Nenkova, A., Vanderwende, L. (2005).** The impact of frequency on summarization. Msr-Tr-2005.

79. **Page, L., Brin, S., Motwani, R., Winograd, T. (1999).** The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab. Previous number = SIDL-WP-1999-0120.

80. **Paice, C. D. (2002).** Review of "Automatic Summarization" by Inderjeet Mani, Amsterdam: John Benjamins (Natural Language Processing Series, Edited by Ruslan Mitkov, Volume 3), 2001, Vol. 28. MIT Press, Cambridge, MA, USA. DOI: 10.1162/coli.2002.28.2.221.

81. **Papineni, K., Roukos, S., Ward, T., Zhu, W.-J. (2002).** BLEU: a method for automatic evaluation of machine translation. Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pp. 311–318.

82. **Parveen, D., Ramsl, H.-M., Strube, M. (2015).** Topical coherence for graph-based extractive summarization. Proceedings of the 2015 conference on empirical methods in natural language processing, pp. 1949–1954.

83. **Parveen, D., Strube, M. (2015).** Integrating importance, non-redundancy and coherence in graph-based extractive summarization. Twenty-Fourth International Joint Conference on Artificial Intelligence, pp. 1298–1304.

84. **Patel, V., Patel, V., Tabrizi, N. (2022).** An automatic text summarization: A systematic review. Computacion y Sistemas, Vol. 26. DOI: 10.13053/cys-26-3-4347.

85. **Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K.,**

Zettlemoyer, L. (2018). Deep contextualized word representations. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, pp. 2227–2237. DOI: 10.18653/v1/N18-1202.

86. **Presser, S. (2020).** Books3. https://twitter.com/theshawwn/status/1320282149329784833.
87. **Radev, D. R., Hovy, E., McKeown, K. (2002).** Introduction to the special issue on summarization. Computational Linguistics.
88. **Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P. J. (2020).** Exploring the limits of transfer learning with a unified text-to-text transformer.
89. **Rajasekaran, A., Varalakshmi, R. (2018).** Review on automatic text summarization. International Journal of Engineering and Technology(UAE), Vol. 7, pp. 456–460. DOI: 10.14419/ijet.v7i2.33.14210.
90. **Ren, P., Chen, Z., Ren, Z., Wei, F., Nie, L., Ma, J., de Rijke, M. (2018).** Sentence relations for extractive summarization with deep neural networks. ACM Trans. Inf. Syst., Vol. 36, No. 4. DOI: 10.1145/3200864.
91. **Rohde, T., Wu, X., Liu, Y. (2021).** Hierarchical learning for generation with long source sequences.
92. **Rush, A. M., Chopra, S., Weston, J. (2015).** A neural attention model for sentence summarization. Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing, pp. 379–389.
93. **Rush, A. M., Chopra, S., Weston, J. (2015).** A neural attention model for abstractive sentence summarization. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, pp. 379–389. DOI: 10.18653/v1/D15-1044.
94. **Sanderson, M., Croft, B. (1999).** Deriving concept hierarchies from text. Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Association for Computing Machinery, New York, NY, USA, pp. 206–213. DOI: 10.1145/312624.312679.
95. **Sandhaus, E. (2008).** The new york times annotated corpus. Linguistic Data Consortium, Philadelphia, Vol. 6, No. 12, pp. e26752.
96. **Saziyabegum, S. (2017).** Review on text summarization evaluation methods. Indian Journal of Computer Science and Engineering, Vol. 8.
97. **See, A., Liu, P. J., Manning, C. D. (2017).** Get to the point: Summarization with pointer-generator networks.
98. **Sharma, E., Li, C., Wang, L. (2020).** BigPatent: A large-scale dataset for abstractive and coherent summarization. ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, pp. 2204–2213.
99. **Sinha, A., Yadav, A., Gahlot, A. (2018).** Extractive text summarization using neural networks. arXiv preprint arXiv:1802.10137.
100. **Song, S., Huang, H., Ruan, T. (2018).** Abstractive text summarization using LSTM-CNN based deep learning. Multimedia Tools and Applications, Vol. 78, pp. 857–875.
101. **Tan, J., Wan, X., Xiao, J. (2017).** Abstractive document summarization with a graph-based attentional neural model. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, pp. 1171–1181. DOI: 10.18653/v1/P17-1108.
102. **Thakare, A. R., Voditel, P. (2022).** Extractive text summarization using LSTM-based

encoder-decoder classification. ECS Transactions, Vol. 107, No. 1, pp. 11665. DOI: 10.1149/10701.11665ecst.

103. Turney, P. (2000). Learning algorithms for keyphrase extraction. Inf. Retr., Vol. 2, pp. 303–336. DOI: 10.1023/A:1009976227802.

104. van Eck, N. J., Waltman, L. (2010). Software survey: VOSviewer, a computer program for bibliometric mapping. Scientometrics, Vol. 84, No. 2. DOI: 10.1007/s11192-009-0146-3.

105. Van Lierde, H., Chow, T. W. (2019). Learning with fuzzy hypergraphs: A topical approach to query-oriented text summarization. Information Sciences, Vol. 496, pp. 212–224. DOI: 10.1016/j.ins.2019.05.020.

106. Vasilyev, O., Dharnidharka, V., Bohannon, J. (2020). Fill in the BLANC: Human-free quality estimation of document summaries. Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems, Association for Computational Linguistics, Online, pp. 11–20. DOI: 10.18653/v1/2020.eval4nlp-1.2.

107. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. (2017). Attention is all you need.

108. Verma, P., Pal, S., Om, H. (2019). A comparative analysis on Hindi and English extractive text summarization. ACM Trans. Asian Low-Resour. Lang. Inf. Process., Vol. 18, No. 3. DOI: 10.1145/3308754.

109. Wang, H., Lu, Y., Zhai, C. (2010). Latent aspect rating analysis on review text data: A rating regression approach. Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, New York, NY, USA, pp. 783–792. DOI: 10.1145/1835804.1835903.

110. Yadav, A. K., Singh, A., Dhiman, M., Kaundal, R., Verma, A., Yadav, D., et al. (2022). Extractive text summarization using deep learning approach. International Journal of Information Technology, pp. 1–9.

111. Yuan, R., Wang, Z., Li, W. (2020). Fact-level extractive summarization with hierarchical graph mask on BERT. arXiv preprint arXiv:2011.09739.

112. Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al. (2020). Big bird: Transformers for longer sequences. Advances in Neural Information Processing Systems, Vol. 33.

113. Zhang, J., Tan, J., Wan, X. (2018). Towards a neural network approach to abstractive multi-document summarization. arXiv preprint arXiv:1804.09010.

114. Zhang, J., Zhao, Y., Saleh, M., Liu, P. J. (2020). PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization.

115. Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., Artzi, Y. (2019). BERTScore: Evaluating text generation with BERT. CoRR, Vol. abs/1904.09675.

116. Zhong, M., Liu, P., Wang, D., Qiu, X., Huang, X. (2019). Searching for effective neural extractive summarization: What works and what's next. CoRR, Vol. abs/1907.03491.

117. Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. The IEEE International Conference on Computer Vision (ICCV), pp. 19–27.

118. Zhuang, L., Jing, F., Zhu, X.-Y. (2006). Movie review mining and summarization. Proceedings of the 15th ACM International Conference on Information and Knowledge Management, Association for Computing Machinery, New York, NY, USA, pp. 43–50. DOI: 10.1145/1183614.1183625.

ISSN 2007-9737

Article received on 02/06/2023; accepted on 10/09/2023.
Corresponding author is Alexander Gelbukh.