# A Comprehensive Survey on Process-Oriented Automatic Text Summarization with Exploration of LLM-Based Methods

Hanlei Jin, Yang Zhang, Dan Meng, Jun Wang\*, Jinghua Tan\* *Member, IEEE* Southwestern University of Finance and Economics, Chengdu, China wangjun1987@swufe.edu.cn, 595915575@qq.com

Abstract—Automatic Text Summarization (ATS), utilizing Natural Language Processing (NLP) algorithms, aims to create concise and accurate summaries, thereby significantly reducing the human effort required in processing large volumes of text. ATS has drawn considerable interest in both academic and industrial circles. Many studies have been conducted in the past to survey ATS methods; however, they generally lack practicality for realworld implementations, as they often categorize previous methods from a theoretical standpoint. Moreover, the advent of Large Language Models (LLMs) has altered conventional ATS methods. In this survey, we aim to 1) provide a comprehensive overview of ATS from a "Process-Oriented Schema" perspective, which is best aligned with real-world implementations; 2) comprehensively review the latest LLM-based ATS works; and 3) deliver an up-todate survey of ATS, bridging the two-year gap in the literature. To the best of our knowledge, this is the first survey to specifically investigate LLM-based ATS methods.

Index Terms—Automatic text summarization, Summarization Survey, Large Language Model (LLM)

#### I. Introduction

ITH the rapid development of World Wide Web, there has been an exponential increase in the volume of textual data, such as news articles, scholarly papers, legal documents, etc. This surge in data has exceeded the capacity of individuals to search and read all the relevant documents. To this end, the field of Automatic Text Summarization (ATS) has emerged as a solution to condense extensive texts into concise and accurate summaries using Natural Language Processing (NLP) based algorithms. ATS enables users to quickly, comprehensively, and accurately understand the core of the original texts. This technique significantly saves time and effort that would otherwise be expended on manually searching, reading, and filtering information.

In the academic research community, scholars often encounter the challenge of navigating through extensive volumes of literature to stay current with developments in their field [1; 2]. This process requires an extensive search and thorough reading of the latest papers. Automatic summarization, as depicted in Figure 1, offers a solution by enabling researchers to efficiently review summaries. ATS allows them to focus on the key content without the necessity of reading each document in its entirety.

ATS is a significant field within NLP, where several past surveys have been conducted. These works often adopt a technical perspective, categorizing ATS methods into "extrac-

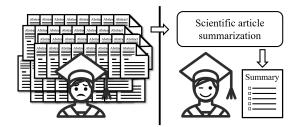
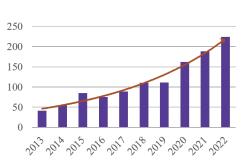


Fig. 1: Example of summarization on academic articles.

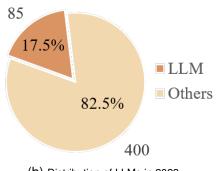
tive" [3; 4], "abstractive" [5; 6], or "hybrid" approaches[7; 8]. Another line of surveys focus on specific domains, such as dialogue[9], biomedical texts[10; 11], news summarization[12; 13], tweet/microblog summarization[14; 15], legal documents summarization[16; 17], scientific papers summarization[18; 19], etc. Nevertheless, current technical-based categorizations (either extractive or abstractive) in ATS [3; 6; 19; 20; 21; 22; 23; 24; 25; 26; 27; 28; 29; 30] may not fully align with practical implementation of ATS algorithms. Furthermore, the advent of Large Language Models (LLMs) may influence ATS methodologies, potentially shifting from the pre-training and fine-tuning paradigm to methods based on prompts.

In this study, our objective is to present a comprehensive survey of Automatic Text Summarization (ATS) techniques using a "Process-Oriented Schema", specifically tailored to align with the practical implementations of ATS applications. In addition, with the advent of Large Language Models (LLMs), we thoroughly investigate and summarize LLM-based ATS methods. To the best of our knowledge, this is the first study to overview the LLM-based ATS methods. The contributions of this study are three-folded, as follows:

- Alignment with Development Pipeline: The study is structured according to the ATS application implementation procedure, utilizing a "Process-Oriented Schema" (defined in Section III). This encompasses the description of Datasets, Pre-processing methods, Modeling Approaches, and Evaluation metrics (as illustrated in Figure 3). Our approach, especially the process-oriented schema, offers a quick reference for users to identify relevant methods according to their development pipeline, effectively serving as a practical ATS guidebook.
- Investigation of LLM-based Approaches: The advent of LLMs has significantly shifted the NLP task paradigms, where the LLMs can serve as a general-purpose language task solver, instead of relaying the conventional pre-training and fine-tuning algorithms. This







2

(b) Distribution of LLMs in 2023

Fig. 2: Overview on the growth and ratio of LLM-based publication in ATS

study delves into LLM-based ATS approaches, establishing itself as one of the earliest works in this area. The study aims to serve as a foundational reference for future ATS research directions in the NLP community.

 Up-to-date Survey: Recognizing the rapid evolution in the field of computer science, we acknowledge that the most recent comprehensive survey on general-purpose ATS was published in 2021 (referenced in Table I). Given the fast-paced advancements in this domain, we emphasize the necessity of continuously updating and revising the methodologies to present an exhaustive, current survey in the field.

This study has conducted a systematic review of extensive journal and conference papers on ATS to offer a thorough overview and a comparative analysis of the ATS process, as well as the applications based in ATS techniques. Furthermore, this review sheds light on the limitations and challenges of existing ATS systems, paving the way for insights into the forefront of today's research and potential future trajectories.

The structure of the subsequent content is organized as follows: We begin by defining the term "Process-Oriented Schema", a novel and implementation-friendly categorization method in Section III. Then, aligning with the intermediate steps of the ATS process, we introduce Data Acquisition in Section IV, and Language/Summarization Modeling in Section VI. Lastly, we discuss the Evaluation metrics in Section VII and ATS based applications in VIII, thereby providing a comprehensive road-map of the entire ATS process.

# II. Background of Automatic Text Summarization

## A. ATS Development Path

The genesis of ATS dates back to the 1950s, notably initiated by Luhn's trailblazing work[31], which emphasized generating summaries through the selection of sentences based on key-word frequencies. Subsequent studies[21; 32] sought to leverage linguistic rules, including word types, grammar, and sentence structure, to interpret the semantic meanings of sentences and consequently generate summaries that captured the essence of the content. However, these initial methods typically required extensive pre-processing[21; 32] and underperformed compared to the nuanced summary generation by

human experts[33]. Consequently, the summaries generated by early ATS systems were criticized for limited semantic depth, lack of coherence, and restricted expressive flexibility[24].

The advent of deep learning models, word embeddings and deep language models has steered the trajectory of ATS towards advanced modeling techniques. Contemporary deep learning models, including Recurrent Neural Networks (RNNs)[34] and Transformers[35], have catalyzed substantial advancements in the ATS field. These methodologies are adept at grasping intricate semantics, thereby producing higher-quality summaries with reduced human efforts.

The emergence of Large Language Models (LLMs) has brought about a myriad of opportunities as well as challenges in the realm of ATS. LLMs have set new benchmarks by significantly improving the accuracy and coherence of generated summaries, outperforming previous approaches. A critical challenge, however, lies in the strategic harnessing of LLMs to emulate the proficiency of expert summary writing. Figure 2 depicts the trend and distribution of publications utilizing LLMs in the ATS field over the past decade.

#### **B.** Related Survey

Many surveys on ATS have been published, they generally adopt a technical categorization perspective, i.e., either classified the ATS methods as "extractive" or "abstractive". Studies such as [23; 24] focused on "extractive" methods, which involve selecting significant sentences or paragraphs from the original documents to compose concise summaries. These works provide an overview of methods based on term frequency, statistical models, or supervised learning approaches. [4] emphasized "extractive" methods on neural models. Compared to extractive methods, "abstractive" summarization involves rephrasing the original text to generate new sentences, where the terms may not directly match those in the original text. Research such as [5; 6] surveyed abstractive summarization methods, tracing the evolution from early statistical and rule-based approaches to recent advancements in neural language models. [26] encapsulated both extractive and abstractive methods.

Another line of surveys focused on domain specific ATS techniques, addressing the challenges presented by various types of content. For instance, [41] offered a comprehensive review of techniques for summarizing multiple documents. In the legal domain, [43; 39; 44] have reviewed specialized

Survey Ref.	Domain	Type	Methods Coverage		Year
[22]	General	Comprehensive	Graph;Machine Learning;Rule-based;Statistical		2009
[23]	General	Extractive	Concept-based;Fuzzy Logic;Graph;Neural Network		2017
[24]	General	Extractive	Graph;Machine Learning;Neural Network;Rule-based;Statistical		2017
[3]	General	Comprehensive	Machine Learning; Neural Network; Rule-based; Statistical; Term Frequency		2018
[6]	General	Abstractive	Deep Language Model; Graph; Reinforcement Learning; Rule-based		2019
[36]	General	Hybrid	Neural Network; Term Frenquency		2019
[27]	General	Abstractive	Deep Language Model;Neural Network;Word Embedding		2021
[26]	General	Comprehensive	Concept-based; Deep Language Model; Graph; Machine Learning; Neural Network;		2021
			Fuzzy Logic;Rule-based;Statistical;Term Frequency;Word Embedding		
[28]	General	Comprehensive	Concept-based;Deep Language Model;Graph;Machine Learning;Neural Network; Fuzzy Logic;Rule-based;Statistical;Term Frequency;Word Embedding		2021
[37]	Domain-Specific	Dialogue	Deep Learning;Knowledge Base;Machine Learning;Retrieval;Term Frequency	774	2017
[38]	Domain-Specific	Dialogue	Machine Learning;Statistical	98	2018
[39]	Domain-Specific	Legal	Graph;Term Frequency;Statistical	166	2019
[20]	Domain-Specific	Graph-based Methods	Graph	348	2019
[40]	Domain-Specific	Scientific Article	Do Not Apply	73	2022

Deep Language Model;Graph;Machine Learning;Neural Network

Do Not Apply

Do Not Apply

TABLE I: Overview of past ATS surveys on their summarization domain, type and methods coverage

methodologies tailored for legal document summarization. [9] focused on summarizing the works on the summarization of dialogue and conversation texts. Additionally, [15] overviewed methods on micro-blog content summarization. Collectively, these surveys contribute to a nuanced understanding of ATS, highlighting the adaptive nature of summarization techniques across varied content domains. We conducted a comprehensive review of highly cited surveys in the field of summarization, leveraging Google Scholar as our primary search platform. Our search strategy involved the utilization of specific keywords, namely "summarization survey" and "automatic summarization survey." The reviewed surveys are organized in Table I. This table categorizes each survey based on its scope, whether it pertains to general summarization methodologies or is specific to a particular domain. Additionally, it provides the techniques covered in each survey, alongside with the their number of citations, and publication year.

Dialogue

Medical

Multi-Document

[9]

[41]

[42]

Domain-Specific

Domain-Specific

Domain-Specific

Despite the valuable insights provided by the aforementioned surveys, they tend to categorize methods primarily from a technical standpoint, which may not directly align with the practical needs of scientists and engineers during specific stages of the summarization development pipeline. In contrast, this study adopts a "Process-Oriented Schema," organizing relevant works in a manner that closely mirrors real-world implementation scenarios. This approach aims to offer enhanced practicality and direct applicability for professionals.

# III. Definition of Automatic Text Summarization Process

The term "Process" has been extensively defined in various professional lexicons. In this context, we selectively reference definitions that resonate most closely with our research framework. The Cambridge Dictionary describes "Process" as "a series of actions that you take in order to achieve a result." Similarly, Collins Dictionary defines it as "a series of

actions which are carried out in order to achieve a particular result." Moreover, Merriam-Webster Dictionary elucidates it as "a series of actions or operations conducing to an end." These definitions collectively underscore the sequential and goal-oriented nature of "Process," a concept central to our study's theoretical foundation.

69

93

2022

2022

2022

Considering that the Automatic Text Summarization techniques are composed of a sequence of intermediate steps aimed at realizing the ultimate goal, i.e., the generation of an abstract for extensive content. We integrate the characteristics of ATS methods with authoritative definitions from dictionaries, to define the term "ATS Process" as the following:

**Definition 1** (Automatic Text Summarization (ATS) Process). A series of automatic actions is performed to distill extensive textual content into concise summaries, aiming to capture the essence of the original text while retaining key information and meaning.

Building upon the definition of ATS, we delineate the intermediate steps necessary to achieve the goal of abstraction generation. The ATS process is illustrated in Figure 3, and its constituent steps are defined as follows:

- Step 1: Data Acquisition. Data Acquisition, the initial step of the ATS process, involves obtaining datasets critical for the system. The specifics of this step are elaborated in Section IV, which provides a comprehensive overview of existing datasets for ATS, along with methodologies for constructing new datasets from scratch.
- Step 2: Text Pre-processing. Pre-processing is a crucial step aimed at refining the collected texts by removing noise and transforming raw texts into a clean, structured format, as elucidated by [23]. This step predominantly employs linguistic techniques such as noise removal, stemming, and sentence/word segmentation to enhance

<sup>&</sup>lt;sup>1</sup>https://dictionary.cambridge.org/dictionary/english/process

<sup>&</sup>lt;sup>2</sup>https://www.collinsdictionary.com/us/dictionary/english/process

<sup>&</sup>lt;sup>3</sup>https://www.merriam-webster.com/dictionary/process

Fig. 3: Process blocks of summarization systems.

the quality of the text data. For a detailed exploration of the pre-processing techniques and their applications in ATS, refer to Section V.

- Step 3: Language/Summarization Modeling. Modeling, the cornerstone of an ATS system, is dedicated to developing versatile language models that can interpret and distill language data into concise summaries. This is achieved through rule-based, statistical, or deep learning approaches which extract patterns from the language data. The process of modeling in ATS is inherently an NLP task, typically commencing with language modeling and subsequently progressing to summarization modeling. The conventional ATS methods are bifurcated into language modeling and summarization modeling. For an in-depth discussion and categorization of the various ATS models, refer to Section VI. This section also provides a detailed exploration of methods based on Large Language Models (LLMs).
- Step 4: Evaluation Metrics. Evaluation metrics of ATS to judge how well ATS works. Objective, comprehensive, and accurate evaluation metrics can lead to the recognition and acceptance of research on ATS. Refer to Section VII for details.

# IV. Data Acquisition

In this study, we summarized that the Data Acquisition process generally employs two approaches:

- Utilization of existing open-source datasets;
- Development of new datasets from scratch.

This section begins by examining open-source datasets commonly employed in prior ATS research. It is important to note that these datasets may not always meet the unique requirements of domain-specific summarization. Additionally, the ATS field often confronts challenges regarding the availability of suitable datasets, as highlighted in several studies[45; 46; 47]. Consequently, we introduce methodologies for dataset creation to address these limitations, providing more extensive data resources to the community and ensuring alignment with specific domain requirements. Moreover, this section outlines strategies for dataset creation that leverage Large Language

Models (LLMs), emphasizing their effectiveness in producing extensive and domain-specific data for ATS applications.

## A. Open-source Datasets

The open-source datasets commonly used in ATS studies are summarized in Table II. The table includes details on the size and domain of each dataset, along with their public accessibility for download. Additionally, we have compiled 3-4 highly cited papers for each dataset to demonstrate the usage of these dataset in research. In the following segments, we overview the datasets listed in Table II.

CNN & Daily Mail[74]: extracts a large news corpus from CNN and Daily Mail websites. The dataset consists of body contents and summaries created with the highlights in the news. Since news editors typically undertake the task of composing these highlights, they inherently serve as a succinct summary of the news content. This corpus has 286,817 training pairs, 13,368 validation pairs and 11,487 test pairs, as defined by their scripts. The source documents in the training set have an average of 766 words spanning 29.74 sentences, while the summaries consist of 53 words and 3.72 sentences on average. Furthermore, it was released in two versions: a non-anonymous version containing the real entity names and an anonymous version with the entity names removed. The anonymous version has a smaller vocabulary, thus early methods often used this version to reduce complexity and improve efficiency[48]. With the development of pre-trained models, handling entity names is no longer difficult, and the non-anonymized version is now commonly used.

**DUC**[75]: DUC (Document Understanding Conferences) is a series of datasets from 2001-2007 released as part of the DUC conference. Each dataset contains 250-1600 news document and summary pairs. There are three forms of the summaries: 1) manually created summaries. 2) automatically created baseline summaries. 3) submitted summaries created by the participating groups' systems. DUC datasets are often used as a test set because they are small but of high quality[6]. Accessing these datasets requires completing several application forms on the website.

**Gigaword**[55]: is a comprehensive archive of news-wire summarization dataset that has been acquired over several

Name	year	Number of pairs	Domain	Used in	Public	URL
CNN & Daily Mail	2016	312,084	News	[48; 49; 50; 51]	<b>√</b>	https://github.com/abisee/cnn-dailymail
DUC 2001-2007	2004	250-1600	News	[52; 53; 48; 54]	X	https://duc.nist.gov/data.html
Gigaword	2003	9,876,086	News	[52; 55; 56; 57]	✓	https://github.com/harvardnlp/sent-summary
XSum	2018	226,711	News	[58; 59; 60; 51]	✓	https://github.com/EdinburghNLP/XSum
Multi-News	2019	56,216	News	[61; 62; 63; 51]	✓	https://github.com/Alex-Fabbri/Multi-News
Scisumm	2019	1,000	Academic paper	[64; 65; 66]	✓	https://cs.stanford.edu/~myasu/projects/scisumm_net/
ArXiv, PubMed	2018	215,000/133,000	Academic paper	[67; 68; 69; 51]	✓	https://github.com/armancohan/long-summarization
WikiHow	2018	230, 843	Knowledge Base	[70; 60; 71; 51]	$\checkmark$	https://github.com/mahnazkoupaee/WikiHow-Dataset
LCSTS	2016	2,400,591	Blogs	[72; 73; 54]	X	http://icrc.hitsz.edu.cn/Article/show/139.html

TABLE II: Open-source datasets for automatic text summarization.

years by the Linguistic Data Consortiume (LDC). It is sourced from seven news media, including Agence France-Presse, Associated Press Worldstream, Central News Agency of Taiwan, Los Angeles Times, Washington Post Newswire Service, Washington Post, Bloomberg Newswire Service, New York Times Newswire Service and Xinhua News Agency. The dataset contains nearly 10 million English news documents and summaries are made up of news headlines. As this data is collectively large, it is suitable for deep neural network training. However, it has been criticized in terms of quality due to its direct use of headlines as summaries[26].

**XSum**[58]: consists of *BBC* news articles and accompanying single-sentence summaries. Specifically, each article is prefaced with an introductory sentence, which is professionally written, typically by the author of the article. The XSum dataset contains 226,711 Wayback archived BBC articles ranging over almost a decade (2010 to 2017) and covering a wide variety of domains (e.g. News, Politics, Sports, Weather, Business, Technology, Science, Health, Family, Education, Entertainment and Arts).

**Multi-News**[61]: consists of news articles and human-written summaries from the site *newser.com*, totaling 56,216 article-summary pairs. Each summary is professionally written by editors and includes links to the original articles cited. Compared to other datasets, it has been demonstrated that the summaries in Multi-News correspond to a lower compression rate and exhibit less variability in the percentage of copied words. Although the Multi-News summaries are notably longer than those in other works, averaging about 260 words, the characteristics enhance the ability of summarization models to generate fluent yet concise text, maintaining coherence throughout its generally longer output.

**Scisumm**[64]: contains the 1,000 most cited academic papers (21,928 citations) in the *ACL Anthology Network*. In addition to the original paper, it provides citation information and summaries annotated by experts in the field. The authors provide a detailed and worthwhile description of the methodology for constructing the dataset, as follows: 1) in the data processing phase, keep the oldest and latest citations and randomly sample the rest so that the 20 citations cover an extended period of time. 2) remove inappropriate citation sentences (i.e., list citations, tables, those with bugs). 3) instead of starting the annotation directly, select 30 articles for five PhD students in NLP to do a preliminary study to verify the validity of the annotation. Experiments show that, by reading the abstract and citing sentences, annotators could

create summaries comparable in quality to the ground truth in an inexpensive way. 4) the final summaries are given by annotators who have read the abstract and citing sentences.

**ArXiv, PubMed**[67]: are two datasets collected from academic repositories, *arXiv.org* and *PubMed.com*. They contain more than 300,000 academic papers in total, with abstracts used as the summaries. This article provides a detailed preprocessing of academic papers. 1) remove the documents that are excessively long (e.g. theses) or too short (e.g. tutorial announcements), or do not have an abstract or discourse structure. 2) remove figures, tables and sections after the conclusion and normalize math formulas and citation markers with special tokens. In addition, special treatment is applied to arXiv: the LATEX files are used and converted to plain text using Pandoc (https://pandoc.org) to preserve the discourse section information.

**WikiHow**[70]: More than 230,000 article-summary pairs are obtained from *WikiHow* knowledge base, which contains online articles describing a procedural task across various topics (from arts, entertainment to computers and electronics). Each article starts with a bold line summarizing the step of a task and is followed by a more detailed explanation. The bold lines at the beginning of each step are extracted and concatenated to form the summaries, while the detailed descriptions are concatenated to form the source article that needs to be summarized. The articles come in two types: one describes a single-method task, while the second represents multiple steps of different methods for a task.

LCSTS[76]: consists of over 2 million real Chinese short blogs from domains such as politics, economics, military, movies, games, etc. Sources include accounts on *Sina Weibo* such as Peopleś Daily, the Economic Observe press, and the Ministry of National Defense. The summaries are annotated manually with rules (e.g. the crawled account has more than 1 million followers) applied to filter out high-quality text. The data was divided into three parts: the first part is the largest amount of master data, the second part is high-quality text pairs obtained by further filtering with manual scoring, and the third part is a more refined test set.

As shown in Table II, the current datasets exhibit the following characteristics: 1) there are large-scale summary datasets available for training deep neural networks and smaller but refined datasets suitable for evaluations. 2) the majority of the datasets are open source and readily accessible. 3) these datasets primarily focus on the news domain. However, it is noteworthy that other domains also require

high-quality datasets, whether in experimental or production environments[26]. For example, as a scholar specializing in a niche area like the summarization of financial earnings releases, customizing a dataset should be requisite. Our paper delineates methodologies pertinent to the construction of such specialized datasets.

## **B.** Building New Datasets

Creating summary datasets typically involves two steps: 1) crawling/fetching texts from websites/databases, and 2) annotate the textual data into summaries. Manual annotation is the most intuitive and traditional practice[75; 61; 64], highly reliable but time-consuming and labor-intensive. In recent years, automatic annotation techniques, including rule-based and LLM-based methods, have been widely used. These methods aim to strike a balance between accuracy and efficiency.

Rule-based annotation: Rule-based annotation typically involves using representative or specific portions of the original text (e.g., title, headline, highlight, first 3 sentences) as summaries based on the structure of the text. News, in particular, is characterized by a special structure, which is why most domains in Table II are related to news. News articles, typically written by journalists, follow the journalistic style, which involves starting with the most important, interesting, or attention-grabbing elements in the opening paragraphs. This structure makes the first few sentences naturally suitable for use as summaries. Similarly, academic papers usually contain abstract sections that naturally serve as summaries. However, the summaries produced based on these rules are often imprecise and condensed. Moreover, texts from other domains do not have a specific format, making them difficult to annotate with rules. At this point, in addition to manual labor, it is more efficient to leverage LLMs.

**LLM-based annotation**: LLMs can be directly employed for the generation of summaries, a process that typically yields satisfactory outcomes. The most straightforward approach is to feed original texts into LLMs to produce summarized outputs, or furthermore stimulate LLMsábility to understand the summarization tasks through prompt engineering or fine-tuning, as elucidated in section VI-C. There are also studies that have designed LLM-based methods specifically for summary annotation. [77] used DialoGPT[78] and transformed summary annotation tasks into keyword extraction, redundancy detection, and topic segmentation tasks to improve the informativeness, relevance, and reduce redundancy of dialog summaries. [79] proposed a GPT-3 based algorithm that annotates medical dialogue data through low-shot learning and ensembles, with a specific focus on capturing medically relevant information. This algorithm outperformed models trained only on human data in terms of both medical accuracy and coherency. If residual concerns persist regarding the quality of these generated results, yet there is a reluctance to invest substantial effort in their evaluation, LLMs may also be utilized to enhance and optimize these outcomes. In this context, [80] proposed an optimisation scheme based on GPT-3 and workers, which brings together the generative strength of LLMs and the evaluative strength of humans.

As a text generation task, text summarization based on

LLMs is also prone to hallucinate unintended text[29], manifesting itself as factually incorrect information. Some studies have been presented to measure and mitigate hallucinated summaries. [81] explored both synthetic and human-labeled data sources for training models to identify word, dependency-, and sentence-level factual errors in summarization. They demonstrated that the best factuality detection model (sentence-factuality model[82] and arc-factuality model[83]) enables training to identify non-factual tokens in the data. [84] proposed generating hard, representative synthetic examples of non-factual summaries through infilling language models to improve factual consistency of datasets. Ensuring the quality of the datasets makes the follow-up work meaningful.

# V. Text Pre-processing

After collecting the data, the next step in the process is pre-processing. Pre-processing is the process of transforming raw text into structured format data. This section describes common methods and powerful tools.

### A. Pre-processing methods

**Noise Removal**: eliminates unnecessary parts of the input text, such as HTML tags in crawled text, extra spaces, blank lines, unknown symbols, and gibberish, etc. In earlier methods, it was necessary to remove stop words from the text [85]. Stop words, commonly occurring words in the text such as articles, pronouns, prepositions, auxiliary verbs, and determiners, were deleted using an artificially designed stop-words file because they were not useful for the analyses [86] and had no significant impact on the selection of summary results.

**Part-Of-Speech (POS)**: involves the assignment of POS tags, such as verbs, nouns, adjectives, etc., to each word in a sentence. This categorization helps in identifying words based on their syntactic roles and context within the sentence structure [87].

**Stemming**: is the process of converting words with the same root or stem to a basic form by eliminating variable endings like "es" and "ed" [88; 89]. This process was initially employed in open vocabulary text mining, for similar reasons as removing stop words: to reduce computation time and enhance recall in information retrieval [90]. [91] proposed reducing each word to its initial letters, a method referred to as Ultrastemming, which demonstrated significant improvements.

**Sentence Segmentation**: splits texts into sentences. The simplest method involves using end markers such as ".", "?", or "!", but it is prone to a lot of interference, including abbreviations like "e.g." or "i.e.". To address this issue, simple heuristics and regular expressions are employed [92]. Sentence segmentation is typically necessary for processing long texts [93] or for specific model designs, such as using sentences as nodes in a graph [94].

Word Tokenization: divides words into subwords. Byte Pair Encoding (BPE) [95] stands out as a simple and highly effective method for subword segmentation. It recursively maps common byte pairs to new ones and subsequently reconstructs the original text using the mapping table. Wordpiece [96] selects the new word unit from all possible options, choosing

7

the one that maximally enhances the likelihood on the training data when incorporated into the model.

## **B.** Pre-process Toolkits

The primary tools for English pre-processing with Python are NLTK<sup>1</sup> and TextBlob<sup>2</sup>. NLTK (Natural Language Toolkit) [97] stands out as a leading platform for developing Python programs that handle human language data. It offers a suite of text processing libraries covering classification, tokenization, stemming, tagging, parsing, and semantic reasoning. NLTK also provides wrappers for industrial-strength NLP libraries and maintains an active discussion forum. TextBlob, built on NLTK, offers a simplified API for various NLP tasks, including part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more. While TextBlob is application-oriented and user-friendly, it sacrifices some flexibility compared to NLTK. For other languages, Jieba<sup>3</sup> and HanLP<sup>4</sup> provide effective solutions for Chinese language. Jieba is a relatively lightweight tool with its primary function being word segmentation, capable of fulfilling most Chinese word splitting needs. On the other hand, HanLP[98] serves as an NLP toolkit built on PyTorch and TensorFlow, contributing to advancing state-of-the-art deep learning techniques in both academia and industry. It has demonstrated notable results in entity segmentation.

Furthermore, diverse models require distinct pre-processing methods as a prerequisite. In earlier models, it was imperative to design and implement an extensive set of rules to eliminate text that posed processing challenges for the model. Technological advancements have led to a noticeable reduction in the labor intensity associated with pre-processing. However, the need for pre-processing remains an indispensable component of the process, albeit with diminished intensity.

# VI. Language/Summarization Modeling

In this section, we categorize ATS modeling into two components: the language model and the summarization model. Automatic Text Summarization (ATS) is a natural language processing (NLP) task that typically involves modeling the language first and then the summarization process. The primary goal of the language model is to transform text into machine-recognizable structured data. Subsequently, the summarization model utilizes this structured data to generate the final summary text.

## A. Language Model

The Language Model (LM) is essential for converting text into structured data. While LM significantly influences text summarization, there is a gap in the literature regarding a thorough exploration of representation methods for Automatic Text Summarization (ATS). This paper addresses this gap by offering a comprehensive summary of LM in the context of ATS. We classify language models into statistical, embedding, and pre-training categories based on their representation

approach. Statistical LMs use statistical features, embedding represents text as continuous vectors, and pre-training dynamically represents text through a deep model.

#### 1) Statistical Language Models

Statistical language models refer to the transformation of text into statistical features such as word frequency, word distribution, position, etc.

**Term Frequency Models**: are often used as a representation method in ATS in the early literature due to their simplicity and efficiency [99; 100; 101; 102]. Specifically, for word i, the TF-IDF is formulated as follows:

$$\mathsf{tfidf}_i = \frac{n_i}{N} \times \log\left(\frac{D}{1+d_i}\right)$$

where  $n_i$  denotes the frequency of word i, N denotes the total number of words, D denotes the total number of documents, and  $d_i$  denotes the number of documents in which word i appears. [103] used the TF-ISF (Term Frequency - Inverse Sentence Frequency) approach to replace documents in the inverse document frequency with sentences, incorporating sentence-level information into the document representation to extract sentences. [104] replaced TF with TW (Term Weights) constructed from word graphs to obtain word contextual relations and obtain a richer word representation. Based on the TF-IDF technique, [105] proposed using the TF-IDF score for each sentence as the weight to be selected to compose the summary. [106] considered that the word distributions may be different across different time frames and hence proposed a Temporal TF-IDF to summarize Twitter posts.

**N-gram Models**: are to form text fragments by sliding a window of size N. N-gram assumes that the occurrence of a word is related to only the preceding and following N words, and the probability of the whole sentence is the product of the probabilities of the individual words. The frequency of all text fragments is counted and filtered by a threshold, and the feature vector of the text is obtained by using the text fragments as features and the frequency as the feature value. [107] directly used N-gram as a representation of sentences. [108], [109] used a bi-gram divided into text fragments of length 2 to indicate N-gram preserves the semantic relationships between words and can improve the preservation of text semantics. The two works [110; 111] leveraged N-gram technique to analyze the characteristics of high-quality summaries.

**Topic Models**: are based on a fundamental assumption that each document is composed of a mixture of topics, and each topic consists of a collection of words. These methods typically represent the text as a matrix containing information about topics and words. Subsequent summarization models often perform classification based on the topic or word matrix. Latent Semantic Analysis (LSA)[112] is an unsupervised technique that captures the semantics of text based on observed word co-occurrences. In the work by [113], a word-sentence matrix was constructed using LSA. This matrix identified relationships between words and sentences through decomposition. Sentences were then filtered by a classifier to generate a summary. Latent Dirichlet Allocation (LDA)[114] is another topic model employed to discover latent topics in a text

<sup>1</sup> https://www.nltk.org/

<sup>&</sup>lt;sup>2</sup>https://github.com/sloria/TextBlob

<sup>&</sup>lt;sup>3</sup>https://github.com/LiveMirror/jieba

<sup>4</sup>https://github.com/hankcs/HanLP

corpus. In the study by [115], LDA was utilized to represent content specificity as a hierarchy of topic vocabulary distributions, yielding state-of-the-art summarization performance. Additionally, [116; 115] applied LDA models to detect the main topics in contents, enabling the selection of top-relevant sentences for composing summaries.

**Pros and Cons:** On the positive side, statistical models come with advantages in simplicity and computational efficiency. However, these models are limited in the extraction of grammatical and contextual relations, owing to their reliance on word frequency rather than syntactic or semantic understanding. Additionally, their performances are sensitive to stop words; the design of the stop word dictionary can significantly affect the performance of the models. Furthermore, statistical models do not inherently construct the "understanding" of the semantics of words, sentences, or documents, limiting their performance. The later word embedding-based approaches are proposed to resolve this issue. Despite their appealing simplicity and efficiency, the limitations of these models underscore the challenges inherent in capturing the richness and complexity of human language.

#### 2) Word Embedding Models

Embedding is a method for converting text into continuous vectors[117]. These continuous vectors can effectively capture the semantic meanings of words while maintaining shorter vector sizes compared to term frequency methods.

Word2Vec[118]: is a word embedding model that employs a shallow two-layer neural network to encapsulate the linguistic context of words in dense, low-dimensional vectors. It functions by simultaneously analyzing individual words and their surrounding context within a sliding window, methodically processing the text corpus to generate meaningful representations. [119; 120] Word2Vec to select the closest sentences to the "centroid" a text based on similarity between embedding vectors, to generate summaries. However, it's noteworthy that Word2Vec may lead to substantial memory consumption, especially with a huge vocabulary size[121]. Moreover, Word2Vec only considers the semantic within a window size, the long distance word-wise dependency might be lost [122].

GloVe (Global Vectors for Word Representation)s[123]: is an unsupervised model that generates word vectors using global word-word co-occurrence statistics from a corpus, skillfully mapping the intricate linear substructures in the word vector space. This mapping adeptly captures semantic relationships, including similarity and analogy between words. Like Word2Vec, GloVe serves as a feature extractor in Automated Text Summarization (ATS) methods. For instance, [124] employed GloVe embedding vectors to identify and aggregate sub-topic related sentences within a text for summary composition. Additionally, [125] fused GloVe with TextRank to align keywords and distill topics with semantic similarity for extractive summarization tasks. While GloVe offers a straightforward and precise word representation, it's not without limitations. The model generates fixed vectors from a comprehensive corpus, which inherently does not address the out-of-vocabulary (OOV) issue. Furthermore, it demands the regeneration of new word vectors when a word

has multiple meanings, a process that can be resource-intensive [122].

**Pros and Cons:** The embedding models are proficient in capturing word representations and learning specific text patterns. Owing to the static nature of the embeddings and the relative simplicity of the neural networks, they are both memory-efficient and expedient in training. However, these methods grapple with challenges in recognizing words that have multiple meanings across various contexts. Additionally, they are limited in capturing deeper text semantics due to the neural networks being relatively shallow, especially when compared to transformer-based[35] architectures. Nevertheless, the issues mentioned have been progressively addressed by subsequent pre-training based approaches[126].

#### 3) Pre-training based Deep Language Models

Pre-training is the process of training on expansive corpora infused with external knowledge to cultivate a universal text representation, has been pivotal in the field of natural language processing[127]. This process enables models to internalize text patterns by fine-tuning the weights within word vectors. Many studies have substantiated the notion that pre-training on extensive corpora can substantially elevate the performance of subsequent tasks[128; 129]. Furthermore, pre-trained models could be directly utilized for ATS, which could save training cost than previous methods[130].

ELMo (Embeddings from Language Models)[131]: use RNN-like encoders as text embeddings. Since the network of RNNs possesses memory cells, it can incorporate information from contextually hidden states into the representation and generation of text at this moment in time, and to generate more informative summaries in conjunction with the preceding context. [132] used ELMo to encode tokens in a corpus of summaries, adding hidden states to the ELMo representations as prior knowledge to unsupervised summary generation. However, due to the limited information that hidden states can "remember", the summaries generated directly by ELMo have difficulty capturing long-range context and cannot guarantee the fluency of the generated text. Therefore, researchers added additional modules for context matching and fluency repair, but this obviously increases the workload.

Transformer Encoder based Models: The Transformerbased encoder used in BERT has demonstrated superior performance in natural language understanding tasks and the generation of word representations enriched with contextual information. Bidirectional Encoder Representation from Transformers (BERT)[128] employs the encoder structure of the Transformer, allowing it to capture deep contextual semantic information. This structure calculates weight coefficients between input words and each word in the preceding sentence, yielding vector representations for the input words[133]. In BERT, the [CLS] token is often used to represent sentences, and sentence scores are obtained by adding classifiers in Automatic Text Summarization (ATS) to select the highest-scoring sentences as summaries. [127] employed a fine-tuned BERT as a sentence representation, significantly outperforming TFbased representations in capturing sentential meaning. [59] introduced a novel document-level encoder based on BERT,

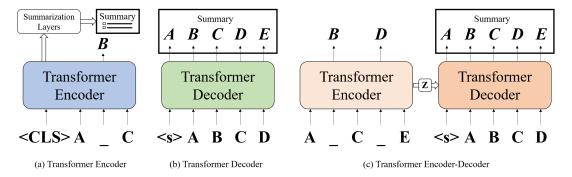


Fig. 4: The base architecture of Pre-trained Transformer for Summarization

replacing the [SEP] token with [CLS] to express document semantics and obtain representations for its sentences. [134] designed HIBERT for summarization tasks, introducing a hierarchical representation concept for multiple documents.

Transformer Decoder based Models: Recent LLMs[129; 135] mostly predominantly adopt a decoder-only architecture, exemplified by GPT (Generative Pre-trained Transformer)[129]. GPT is a multi-layer Transformer decoder that undergoes pre-training on a large corpus of text, followed by fine-tuning for downstream tasks. GPT is considered to have a auto-regressive nature, by continuously predicting the next word given the preceding words in a context. Decoderonly model demonstrate superior performances when trained on extensive text data across diverse tasks. They also demonstrate robust summarization capabilities and mitigate expressiveness degradation caused by the low-rank problem in the encoder attention matrix[136]. Section VI-C will delve into summarization based on LLMs which are constructed based on the decoder-only architecture.

Transformer Encoder-Decoder based Models: Prior to the emergence of LLMs, the Transformer encoder-decoder based models served as the cornerstone of generative summarization. Google's Pegasus[60] is a pre-training model tailored for summarization. Pegasus adopts a unique approach by removing or masking sentences from the input document and utilizing the remaining sentences to generate the masked ones in the output. [137] leverage Pegasus to delve into highly-abstractive multidocument summarization, where the summary is explicitly conditioned on a user-provided topic statement or question. BART[138] is specifically designed for sequence-to-sequence tasks, encompassing text summarization. The model follows a two-step process: 1) defining an arbitrary noising function to corrupt the original sentence, and 2) training BART to reconstruct the original text. This unique approach combines the strengths of both auto-encoder and auto-regressive pretraining objectives, enhancing its understanding and generation of texts. T5[139] conducts a comprehensive exploration of various language understanding tasks and establishes that the encoder-decoder architecture with the denoising objective excels in summarization tasks.

Figure 4 illustrates the details of various Transformer architectures applied to Automatic Text Summarization (ATS). The process of acquiring a pre-trained model in this manner is tailored to the specific downstream summarization task. A well-designed pre-training strategy, customized for ATS, can

markedly enhance the performance of the overall summarization task. Pre-trained models could be directly adopted for summary generation, which could help to reduce the cost to train new models for different tasks and datasets.

**Pros and Cons:** Pre-training based deep language models excel in capturing profound text semantics through their complex neural architecture, resulting in performances that surpass those of earlier methods based on statistics and shallow neural networks. Nonetheless, these models require task-specific modifications to the neural architecture to adapt to the summarization task, in addition to fine-tuning during training. This leads to increased training costs.

#### **B. Summarization Model**

Summarization models use the structured data constructed by language models (LM) to produce the final summary text. In previous studies, summarization models are usually classified into extractive and abstractive, which is followed in this paper. Extractive summarization models tend to select top-k sentences to combine into a summary based on importance. Abstractive summarization models predict the token with the highest probability to generate a summary in a text generation manner. With the recent rise of LLMs, there has been a heightened interest in leveraging LLM-based methods for ATS[140], hence, this paper further explores LLM-based summarization.

#### 1) Extractive Summarization

Extractive summarization models usually select sentences from the original text for the summary, which can be divided into the following three steps: 1) calculate the importance of each sentence, 2) sort the sentences according to their importance, 3) select top-k sentences as the summary. Since extractive summarization requires the selection of important sentences, these approaches can be categorized according to different definitions of importance.

Words-counting Models: define importance as "most frequent" or "most likely to occur" [23], for example, selecting the sentence with the highest frequency of words as the summary. More studies optimize on the basis of word frequency. TextRank[141] is an unsupervised ATS method for extracting the most important keywords and sentences from documents. It estimates the degree of similarity between two phrases based on the elements they contain. This overlap is calculated as the number of shared lexical tokens divided by the length of each sentence. LexRank[142] is a probabilistic technique for determining sentence importance, which is based on the concept of

feature vector centrality for phrase graph representation. It uses a connectivity matrix based on intra-sentence cosine similarity, which is employed as an adjacency matrix in sentence graph representations. [143] proposed using the amount of concepts contained in sentences as a score. Words-counting models require less processor capacity and memory, without any extra linguistic knowledge or complex linguistic processing, and are widely used in production scenarios. However, they receive a lot of interference from irrelevant words, resulting in high scoring sentences but were not important.

Similarity-based Models: In contrast to simple quantitative statistical features, similarity-based models assign high ranks based on similarity. Clustering is a relatively quick measure of similarity that identifies the most central sentences, ensuring they cover the relevant and important information for the members of the cluster. [100] proposed a method that clusters at the word level. It filters out the central word, and then the sentence containing the most central words is selected as the summary. [144] assumed that when a sentence has more similar sentences, it will be considered more important or more representative. The model clusters and ranks sentences after converting sentences into vectors, and the sentence closest to the cluster center is selected as the summary. Thus, the method of converting sentences into vectors plays a crucial role. [120] applied word2vec for feature extraction and used K-Means for sentence summarization. [145] constructed the word vectors by GloVe and achieved better results than TF-IDF. Similarity-based models enhance word and sentence coherence and reduce interference from unrelated texts but may fail to identify semantically equivalent sentences.

Classifiers: typically operate as follows: 1) cut and represent the sentences, 2) categorize the sentences using a classifier, 3) select the sentences whose category is summary. Thus, most classification models can be adapted to the summarization task. [146] proposed SqueezeBERTSum, a trained summarization model fine-tuned with the SqueezeBERT encoder variant. [147] investigated generative and discriminative training techniques to fuse domain knowledge into knowledge adapters and applied adapter fusion to efficiently inject the knowledge adapters into the basic pre-training models for fine-tuning the summarization task. These approaches allow previous better performing classification models to be applied to summarization, but they require a large data set of summaries such that each sentence labeled as either summary or non-summary, which is not customary for summary generation.

**Pros and Cons:** Extractive summarization models, by directly extracting sentences from the original texts, demonstrate an enhanced capability to capture precise terminologies, thereby achieving heightened accuracy. Furthermore, these models exhibit a distinct advantage in terms of lower requirements for extensive training data, rendering them a cost-effective and expeditious option. Despite these advantages, it is essential to acknowledge that extractive models diverge significantly from the approach employed by human experts in summary generation. They may encounter limitations related to expressive quality, leading to summaries that fall short of the rich and nuanced content produced by human experts. The extracted sentences, while accurate, may exhibit

redundancy, excessive length, and contextual contradictions, thereby presenting challenges in producing summaries that align seamlessly with human-generated counterparts.

#### 2) Abstractive Summarization

Abstractive summarization models generate summaries in the form of text generation, producing sentences that may differ from the original text. Summarization with the abstractive approach is naturally a sequence-to-sequence task, making sequence-generated models applicable to ATS. With the development of equipment and technology, abstractive methods have been able to achieve higher quality and are becoming the mainstream. This paper focuses on the modeling of RNN and Transformer based Models.

RNN-based Models: include the ontological Recurrent Neural Network(RNN)[34] and RNN's variants such as Long Short-Term Memory (LSTM)[148] and Gated Recurrent Unit (GRU) [149], which convert the input sequence of a neural network into a sequence of letters, words or sentences using seq2seq. SummaRuNNer[150] is an interpretable RNN-based model for extractive document summarization that allows intuitive visualization, and it has been shown to perform better than or be comparable to state-of-the-art deep learning models. [151] introduced a conditional RNN that generates a summary of an input sentence using a novel convolutional attentionbased encoder. This encoder ensures that the decoder However, as the length of the sequence increases, problems such as gradient disappearance and gradient explosion start to affect their performance[152]. To address these problems, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) were proposed. LSTM introduces a gating mechanism compared to earlier RNNs, which retains critical information and eliminates useless information by controlling the flow of information. [153] compared the impact of the local attention in an LSTM model to generate summaries and showed that the global attention-based LSTM model produces better. The GRU is a further simplification of the LSTM, with only two gates (reset and update), and is less computationally intensive and faster to train than the LSTM. [154] proposed a two-stage structure, the first of which is a key-sentence extraction, followed by the GRU-based model to handle the extractive summarization of documents. RNN-based models demonstrate proficiency in producing coherent summaries aligned with human writing patterns owing to their sequential computational architecture. Nevertheless, the controllability of generation quality is not consistently assured, and there exists a potential for sluggish computational performance.

**Transformer-based Models**: have made significant progress, especially in the presence of pre-training. After fine-tuning on the summarization datasets, pre-training models introduced in VI-A3 such as BART and GPT can be directly applied to generate summaries[53]. Additionally, there is ongoing work on improving Transformers for summary tasks. Part of the work focuses on improving the Transformer's ability to summarize long text. [155] introduced a novel hierarchical propagation layer where the input is divided into multiple blocks independently processed by the scaled dot-attentions. These blocks are then combined between

successive layers to spread information between multiple Transformer windows. [156] assumed a hierarchical latent structure of a long document where the top-level captures the long range dependency at a coarser time scale, and the bottom token level preserves the details. They then captured the two representations with Transformers. There are also topic-specific and differently focused summarization studies. [157] integrated topic models into Transformer and clarified document semantics and improving results without breaking the structure of the Transformer. Socratic pre-training[158] trains Transformers to generate and answer relevant questions in a given context, enabling the model to more effectively adhere to user-provided queries and identify relevant content to be summarized. Transformer-based models do not require sequential computation and can process the entire sequence simultaneously. In the summary task, these models can take the whole sentence as input and calculate the importance of each word in the sentence to better understand the semantics.

**Pros and Cons:** The abstractive summarization models generates summary in a way that more closely resembles humans by the next token prediction. Compared to the extractive approach, this approach is characterized by flexibility in expression and commendable compression ratio. Nevertheless, it is imperative to acknowledge the inherent challenges associated with the development of this approach. The complexity of implementing this technique is notably high, often demanding datasets of superior quality for effective training. Additionally, the utilization of this method entails a substantial consumption of computational resources and training time, requiring a tradeoff between cost and efficiency.

## C. Large Language Model (LLM) based ATS

Large Language Models (LLMs) refer to Transformer language models that contain billions (and more) of parameters, which are trained on massive text data and various tasks[159]. Most LLMs use an auto-regressive structure similar to GPT, possessing the ability to transfer to the ATS[160]. LLMs have shown promising results across a wide range of domains, including summarization, question answering, mathematics, logic inference, etc.[159]. Some studies evaluating the use of LLMs in ATS have demonstrated effects surpassing human performance[161; 162; 163; 164; 165]. Despite major stylistic differences, such as the amount of paraphrasing, [163] found that LLMs-generated summaries are judged to be on par with human written summaries. [166] showed that not only do humans overwhelmingly prefer summaries generated by LLMs, prompted using only a task description, but these summaries also do not suffer from common dataset-specific issues such as poor factuality. There is still a dearth of surveys on the use of LLMs in ATS, and our article will sort that out.

To the best of our knowledge, the use of LLMs for summarization was first developed by OpenAI, updating the entire GPT-3 model with reinforcement learning algorithms to enable the LLMs to generate summaries that match human preferences[167]. Subsequent work has followed a similar path[168]. However, as model sizes increase, full parameter training of LLMs becomes costly. Research on full parameter training is gradually decreasing in favor of more cost-effective

and efficient approaches, including knowledge distillation, fine-tuning and prompt engineering.

#### 1) Knowledge Distillation from LLMs

Basically, a knowledge distillation system is composed of three key components: knowledge, distillation algorithm, and teacher-student architecture[169]. In knowledge distillation, a small student model is generally supervised by a large teacher model[170]. Through knowledge distillation, SREFEREE[171] starts with GPT3-generated summaries and trains smaller but better summarizers with sharper controllability. [172] proposed a versatile and compact summarization model derived from GPT-3.5 through distillation.

#### 2) Fine-tuning on LLMs

Given the huge volume of LLMs, the parameter structure of LLMs is usually not updating directly, but fine-tuned. [173] utilized an efficient few-shot method based on adapters, pre-trained the adapters on a large corpus, and then fine-tuned them on the small available human-annotated dataset. [174] fine-tuned several state-of-the-art Transformer models on a newly created medical dialogue dataset and found that models pre-trained on general dialogues outperform baseline model. There is also work with prompt tuning and prefix tuning[175; 176]. LLMs for summary generation are mostly based on the prompt engineering methodology, from which they investigate the most effect prompt words to best stimulate the summary generation ability of LLMs.

#### 3) Prompt Engineering for LLMs

Prompt engineering explores the strategic formulation of prompts to maximize the exploitation of specific capabilities inherent in Large Language Models (LLMs). This process entails adapting the original input text string, x, into a refined version, x', to more effectively draw upon the LLMs' inherent knowledge and thereby enhance the interpretation of the input text [177]. This, in turn, significantly uplifts the quality of the generated summaries. Notably, prompt engineering is advantageous as it obviates the need for extensive training or relies on merely a small set of samples [178], thereby offering a reduction in resource expenditure. This study aims to detail various implementations of prompt engineering within the domain of Automated Text Summarization (ATS), with a particular focus on methodologies such as template engineering, chain of thought (CoT), and agent interactions.

**Template Engineering**: The most natural way to create prompts is to manually create intuitive templates based on human introspection. Here are some examples of manual templates:

- [Input] TL;DR: [177]
- Article: [Input]. Summarize the article in three sentences.
   Summary: [163]
- Please generate a one-sentence summary for the given document. [Input] Try your best to summarize the main content of the given document. And generate a short summary in 1 sentence for it. Summary: [159]

While the strategy of manually crafting templates is intuitive and does allow solving various tasks with some degree of

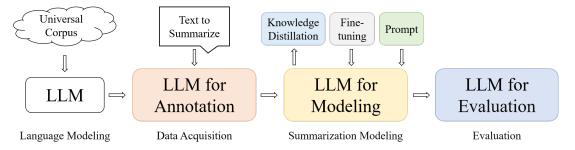


Fig. 5: Process of automatic text summarization achieved by LLM

accuracy, there are also several issues with this approach: 1) Creating and experimenting with these prompts is an art that takes time and experience, particularly for some complicated tasks such as semantic parsing[179]. 2) Even experienced prompt designers may fail to manually discover optimal prompts[180]. To address these problems, several methods have been proposed to automate the template design process. [178] prompted target summaries with entity chains—ordered sequences of entities mentioned in the summary. [181] obtained summaries by manually designing templates and retrieving relevant information from source documents or what the authors call *retrieve-then-summarize* method.

Chain of Thought: Chain of thought is a series of intermediate reasoning steps that significantly improves the ability of LLMs to perform complex reasoning[182]. To address factual hallucination and information redundancy in ATS, [183] proposed a Summary Chain-of-Thought (SumCoT) technique to elicit LLMs to generate summaries step by step, which helps them integrate more fine-grained details of source documents into the final summaries, correlating with the human writing mindset. [184] designed "Chain of Density" (CoD) to generate detailed and entity-centric summaries. The experiments showed that summaries generated by CoD are more abstractive, exhibit more fusion, and have less of a lead bias than GPT-4 summaries generated by a vanilla prompt.

**Agent Interactions**: Agents are artificial entities that sense their environment, make decisions, and take actions[185]. [186] proposed a tri-agent generation pipeline consisting of a generator, an instructor, and an editor to enhance the customization of generated summaries of LLM to better fulfill user expectations.

In fact, we observe that LLM can already function as a self-contained process, demonstrating deep involvement and yielding superior results. Figure 5 illustrates the ATS process achieved by LLMs. Given that LLMs have achieved satisfactory performance in summarization tasks, surpassing even the benchmark of reference summarization, some scholars argue that most routine work in the field of text summarization is no longer necessary in the era of LLMs[187]. However, we believe that there is still value in traditional models, especially in situations where privacy[188] and data security need assurance, as well as in some low-resource scenarios.

**Pros and Cons:** Large Language Models (LLMs) have significantly advanced NLP tasks by minimizing training expenses, enabling summarization through streamlined prompts or Chain of Thought (CoT) methods. Nonetheless, LLMs present multiple challenges. First, their outputs can be in-

consistent, leading to varying performance levels. Second, alterations in prompt wording can substantially affect the quality of summarization, necessitating additional research into prompting techniques. Finally, some domain-specific scenarios that demand supplementary training can result in significantly increased costs compared to the previous methods.

# VII. Evaluation

Evaluation is the process of exemplifying the quality of summaries[189]. Stable and consistent assessment metrics are essential for summarization works. Previous studies have explored the assessment of summarization quality. [190] proposed that summaries should be measured in a multi-dimensional way, including relevance, factual consistency, conciseness and semantic coherence. Relevance refers to whether the candidate summary contains the main ideas.[101] considered that summarization can be measured in terms of redundancy, relevance and informativeness. According to these studies, our paper categorizes summary evaluation methods into three categories, including overlap, similarity and LLM-based evaluation metrics.

#### A. Metrics based on Term Overlap

Overlap is the most widely used method for evaluating summarization. It efficiently considers the matching of words between candidate summaries C and reference summaries S. Through word matching, metrics such as precision P, recall R and F-score F can reflect the overlap[22].

$$P = \frac{|S \cap C|}{C}, R = \frac{|S \cap C|}{S}, F = \frac{(\beta + 1) \times P \times R}{\beta^2 \times P + R}$$

$$s.t. \ \beta^2 = \frac{1 - \alpha}{\alpha}, \alpha \in [0, 1]$$
(1)

Individual word overlap methods are less frequently used because they do not take into account the order of the candidate summaries. There have been improvements to these metrics.

**ROUGE** (Recall-Oriented Understudy for Gisting Evaluation)[191]: is the most common tool for ATS evaluation. It includes measures to count the number of overlapping units such as n-gram, word sequences, and word pairs, through ROUGE-N, ROUGE-L, ROUGE-W and ROUGE-S.

ROUGE-N is based on the n-gram as the basic unit to calculate the n-gram overlap between C and S. Formally, ROUGE-N is computed as follows:

$$ROUGE-N = \frac{\sum_{S} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S} \sum_{gram_n \in S} Count(gram_n)}$$
 (2)

Where n stands for the length of the n-gram, which is a hyperparameter, typically set to n = 1/2/3.  $qram_n$  is the maximum number of n-grams co-occurring in C and S. The intuition of ROUGE-L is that the longer the longest common subsequence (LCS) of two summary sentences is, the more similar the two summaries are. Similar to Equation 1, dividing the LCS length by the length of C or S gives precision Pand recall R of ROUGE-L, and further calculates F-score F. ROUGE-W weights the score on word position to compensate for the inability of the LCS to distinguish spatial relations within sequences. ROUGE-S uses skip-bigram co-occurrence instead of n-gram, allowing for arbitrary gaps. The stability and reliability of ROUGE at different sample sizes were reported by the author in [192]. However, since ROUGE focuses on recall, it does not directly consider the fluency and conciseness of a summary[30].

**BLEU** (Bilingual Evaluation Understudy)[193]: Compared to ROUGE, the difference is that BLEU uses the precision calculation.

$$BLEU-P = \frac{\sum_{C} \sum_{gram_n \in C} Count_{clip}(gram_n)}{\sum_{C} \sum_{gram_n \in C} Count(gram_n)}$$
 (3)

Note that intersection of multi-sets captures the notion of clipped counts, i.e. upper bounding of the total count of each candidate word by its maximum reference count. In contrast to recall, precision penalizes sentences that are too long, and thus, shorter sentences tend to get high scores. Therefore, a brevity penalty was introduced to penalize texts shorter than the length of a reference:

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{1-r/c} & \text{if } c \le r \end{cases} \tag{4}$$

Then,

$$BLEU = BP \cdot exp(\sum_{n=1}^{N} w_n log BLEU - P)$$
 (5)

where N is the maximal n-gram order, and  $w_n$  is n-gram weight, set to 1/N. Similar to the issues with ROUGE, BLEU focuses on the precision rate, which only considers the degree of overlapping, ignoring the diversity of grammar and expression.

NIST (National Institute of standards and Technology)[194]: is an improvement on the BLEU method. While the BLEU algorithm simply adds up the number of n-grams, NIST adds up the information and divides it by the number of n-grams. This is equivalent to giving more weight to words that have less emphasis.

METEOR (Metric for Evaluation of Translation with Explicit ORdering) [195]: was designed to explicitly address the weaknesses in BLEU. It uses WordNet to treat spelling variants, synsets, and paraphrase tables, and it distinguishes between function and content words.

**PPL** (**Perplexity**): focuses on estimating the probability of a sentence occurring based on each word and normalizing it with the sentence length. Abstractive models are able to output the probability of the next word and can apply PPL simply and quickly. However, PPL is determined by self-generated probabilities and lacks objectivity.

## **B.** Metrics based on Similarity

Similarity-based metrics measure quality by calculating the similarity between candidate summaries and reference summaries. BERTSCORE[196] computes a similarity score for each token in the candidate sentence with each token in the reference sentence and correlates better with human judgments, providing stronger model selection performance. [197] showed that the max cosine similarity value over each dimension of the summary ELMo word embeddings is a good representation that results in high correlation with human ratings. However, the similarity-based approach ignores measures such as fluency and factuality of summary results.

#### C. Metrics based on LLMs

While the above metrics do not yet cover all aspects of summary quality assessment, LLMs provide a variety of perspectives and assess summarization quality from a human mind. [198] explored ChatGPT's ability to perform humanlike summarization evaluation using four human evaluation methods including Likert scale scoring, pairwise comparison, Pyramid, and binary factuality evaluation. By designing prompts that allow ChatGPT to answer ratings, preferences, etc. for summary results, it outperformed commonly used automatic evaluation metrics on some datasets. [199] provided GPT-3 with a small number of in-context examples to score for consistency, relevance, fluency and coherence. [200] proposed to model objective and subjective dimensions of generated text based on role-players prompting mechanism and generate final evaluation results based on the synthesis of judgments from multiple roles. Experimental results showed high consistency with human annotators. For unsupervised evaluation of book-length texts (¿100K tokens), [201] presented the BOOOOKSCORE, evaluation scores were obtained by having the GPT-4 answer whether the generated results had causal omission, salience, duplication possible and etc. problems.

In addition to comprehensive evaluation metrics, there are a considerable number of studies that evaluate the faithfulness and factuality of summary results based on LLMs. [202] particularly explored ChatGPT's ability to evaluate factual inconsistency under a zero-shot setting by examining it on both coarse-grained and fine-grained evaluation tasks, including binary entailment inference, summary ranking, and consistency rating. The study proved that ChatGPT generally outperformed previous evaluation metrics, indicating its great potential for factual inconsistency evaluation. [203] proposed a new benchmark called FIB (Factual Inconsistency Benchmark) that focuses on the task of summarization. The study found that existing LLMs generally assign a higher score to factually consistent summaries than to factually inconsistent summaries. [204] designed TrueTeacher, a method for generating synthetic data by annotating diverse model-generated summaries using an LLM to predict the corresponding factual consistency. [205] introduced FFLM, a zero-shot faithfulness evaluation that combines probability changes. Experiments showed that FFLM performed competitively with or even outperformed ChatGPT on both inconsistency detection and faithfulness rating with fewer parameters.

As the understanding and generative capacity of LLMs continue to advance and evolve, the development of assessment metrics predicated based on the capabilities of these models is anticipated to yield results that are progressively more objective, equitable, and varied in scope and application.

# **VIII.** ATS based Applications

The ultimate objective of Automated Text Summarization (ATS) is to enhance the efficiency of information retrieval and analysis in real-world scenarios. ATS has applications in various domains, including the summarization of news articles, scientific papers and other areas that involve substantial reading efforts.

News Summarization: represents the most extensively investigated domain within ATS, largely owing to the richness and maturity of available datasets, coupled with the significant demand for its application across various industries. [206] introduced sentence fusion, a text-to-text generation technique that merges phrases containing similar information into a single sentence for summary generation. Due to the real-time nature of news, [207] presented Newsblaster, a system that is based a clutering meethods for crawled news articles by topic, and thus produce summaries based on topic clusters. In addition, based on the chronological nature of the forward and backward relations of news articles, [208] proposes a time-line based summarization method,

**Novel Summarization**: focuses on condensing novel articles, texts characterized by their extended length. [209] suggested calculating the alignment score using ROUGE and METEOR metrics to identify the most suitable sentences for crafting summaries. [209] developed two techniques for creating fictional character descriptions, deriving from the articles' attributes and centered on the dependency relationships among pivotal terms.

Scientific Paper Summarization: presents challenges including managing citation relationships, specialized structural parsing, and generating accurate proper names. CGSum[210] is a model for summarization based on citation graphs, capable of integrating information from both the primary paper and its references using graph attention networks[211]. SAPGraph [212] offers an extractive summarization framework for scientific papers, utilizing a structure-aware heterogeneous graph. This framework represents the document as a graph with three types of nodes and edges, drawing on structural information from facets and knowledge to model the document effectively.

Blog Summarization: is crucial for accessing real-time information, with platforms like Twitter and Facebook hosting millions of highly pertinent and timely messages for users. [213] introduced a participant-based event summarization approach that expands the Twitter event stream through a mixture model, identifying significant sub-events tied to individual participants. [214] formulated a methodology to globally and locally model temporal context, proposing an innovative unsupervised summarization framework that incorporates social-temporal context for Twitter data. [214] examined various algorithms for the extractive summarization of microblog posts, presenting two algorithms that generate summaries by selecting a subset of posts from a specified set.

Dialogue Summarization, encompassing summarization of meetings, chats, and emails, is an increasingly pertinent task to deliver condensed information to users. [215] identified that the challenge in dialogue summarization lies in the heterogeneity arising from multiple participants' varied language styles and roles. [216] introduced a query-based dialogue summarization system that selects and extracts conversational discourse based on the overall content and specific phrase query information. [217] developed a Zero-shot approach to conversation summarization, employing discourse relations to structure the conversation and leveraging an existing document summarization model to craft the final summary.

Medical Summarization: Summarization in the medical field holds substantial clinical significance, as it has the potential to expedite departmental workflows (e.g., in radiology), diminish redundant human labor, and enhance clinical communication [218]. [219] crafted a universal framework for evaluating the factual accuracy of summaries, utilizing an information extraction module to conduct automated fact-checking on the citations within generated summaries. [220] devised a novel implicit-based text exploration system, tailoring it to the healthcare sector. This system allows users to navigate the results of their queries more deeply by traversing from one proposition to another, guided by a network of implicit relations outlined in an implicit graph. [221] introduced a model founded on seq2seq that integrates essential clinical terms into the summarization process, demonstrating marked improvements in the MIMIC-CXR open clinical dataset.

#### IX. Conclusion

In this survey, we offer an exhaustive review of Automatic Text Summarization (ATS) methods, employing a "Process-Oriented Schema" that resonates with the practical application of ATS techniques. We aim to make three contributions by this study: 1) propose a "Process-Oriented Schema" survey perspective for ATS, which is best aligned with real-world implementations; 2) comprehensively review the latest LLM-based ATS works, which is one of the earliest word within ATS; and 3) deliver an up-to-date survey of ATS, bridging the two-year gap in the literature.

This survey is segmented into four main parts from the definition of "Process-Oriented Schema": Data Acquisition, Text Pre-processing, Language/Summarization Modeling, and Evaluation Metrics. Regarding datasets, our survey introduces and contrasts the commonly utilized datasets, offering strategies for crafting new datasets from the ground up, particularly those optimized for LLMs to enhance dataset quality. We proceed to delineate techniques for dataset pre-processing and introduce robust toolkits to streamline their utilization. Subsequently, we bifurcate the model into two segments: language modeling and summarization modeling. In the language modeling segment, we spotlight pre-training strategies that significantly bolster downstream tasks. As for summarization models, we review seminal works and dissect the pros and cons of both extractive and abstractive methodologies. We also collected and analyzed extensive LLM-based methods, in comparison with the conventional methods. For Post-modeling, we introduce established evaluation metrics and probe into the role of LLMs

in assessing the depth and accuracy of summaries. At the end, we discussed the ATS based applications.

In considering the future directions of ATS, we recognize that LLMs bring overwhelming advantages over traditional methods, notably in the quality and flexibility of the generated texts, and the prompting paradigm to alleviate the cost of training deep models. While many ATS tasks could be directly addressed using LLMs, this also opens up potential new research directions: 1) The effectiveness of LLM-based summarization could be significantly influenced by the composition of prompt words, hence, devising strategies for effectively designing prompts could become a crucial direction for LLM-based ATS; 2) Given that LLMs are trained with generalized knowledge, formulating efficient training strategies to align with domainspecific ATS tasks with low training resources, such as those in medical, news, and scholarly domains, presents another valuable research direction; 3) Considering the inconsistency in text outputs, another critical area of exploration involves evaluating LLM-based ATS outputs and ultimately stabilizing the generation process.

# Acknowledgments

This work has been supported by grants awarded to Prof. Dan Meng from the Education and Teaching Reform Project for the Central Universities (Southwestern University of Finance and Economics) [No. 2023YJG043], and Jiaozi Institute of Fintech Innovation, Southwestern University of Finance and Economics[No. kjcgzh20230202].

#### References

- [1] N. Agarwal, K. Gvr, R. S. Reddy, and C. P. Rosé, "Scisumm: a multi-document summarization system for scientific articles," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics: Human Lang. Technol.: Syst.Demonstrations*, ser. HLT '11. USA: Assoc. Comput. Linguistics, 2011, p. 115–120.
- [2] G. Lev, M. Shmueli-Scheuer, J. Herzig, A. Jerbi, and D. Konopnicki, "Talksumm: A dataset and scalable annotation method for scientific paper summarization based on conference talks," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds. Florence, Italy: Assoc. Comput. Linguistics, Jul. 2019, pp. 2125–2131.
- [3] N. Nazari and M. A. Mahdavi, "A survey on automatic text summarization," J. AI and Data Mining, vol. 7, no. 1, pp. 121–135, 2019.
- [4] D. Suleiman and A. Awajan, "Deep learning based abstractive text summarization: Approaches, datasets, evaluation measures, and challenges," *Math. Problems Eng.*, pp. 1–29, 2020.
- [5] S. Gupta and S. K. Gupta, "Abstractive summarization: An overview of the state of the art," *Expert Syst. Appl.*, vol. 121, pp. 49–65, 2019.
- [6] H. Lin and V. Ng, "Abstractive summarization: A survey of the state of the art," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 01, pp. 9815–9822, Jul. 2019.
- [7] I. K. Bhat, M. Mohd, and R. Hashmy, "Sumitup: A hybrid single-document text summarizer," in *Soft Comput.: Theories and Appl.*, M. Pant, K. Ray, T. K. Sharma, S. Rawat, and A. Bandyopadhyay, Eds. Singap.: Springer Singap., 2018, pp. 619–634.
- [8] E. Lloret, M. T. Romá-Ferri, and M. Palomar, "Compendium: A text summarization system for generating abstracts of research papers," *Data & Knowl. Eng.*, vol. 88, pp. 164–175, 2013.
- [9] X. Feng, X. Feng, and B. Qin, "A survey on dialogue summarization: Recent advances and new frontiers," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, *IJCAI 2022*, *Vienna, Austria, 23-29 July 2022*, L. D. Raedt, Ed. ijcai.org, 2022, pp. 5453–5460.
- [10] H. D. Menéndez, L. Plaza, and D. Camacho, "Combining graph connectivity and genetic clustering to improve biomedical summarization," in 2014 IEEE Congr. Evol. Comput. (CEC), 2014, pp. 2740–2747.

- [11] A. Chaves, C. Y. Kesiku, and B. Garcia-Zapirain, "Automatic text summarization of biomedical text data: A systematic review," *Inf.*, vol. 13, p. 393, 2022.
- [12] A. Sahni and S. Palwe, "Topic modeling on online news extraction," in *Intell. Comput. and Inf. and Commun.*, S. Bhalla, V. Bhateja, A. A. Chandavale, A. S. Hiwale, and S. C. Satapathy, Eds. Singap.: Springer Singap., 2018, pp. 611–622.
- [13] P. Sethi, S. Sonawane, S. Khanwalker, and R. B. Keskar, "Automatic text summarization of news articles," in 2017 Int. Conf. Big Data, IoT and Data Sci. (BID), 2017, pp. 23–29.
- [14] N. Vijay Kumar and M. Janga Reddy, "Factual instance tweet summarization and opinion analysis of sport competition," in *Soft Comput. and Signal Process.*, J. Wang, G. R. M. Reddy, V. K. Prasad, and V. S. Reddy, Eds. Singap.: Springer Singap., 2019, pp. 153–162.
- [15] S. Dutta, V. Chandra, K. Mehra, S. Ghatak, A. K. Das, and S. Ghosh, "Summarizing microblogs during emergency events: A comparison of extractive summarization algorithms," in *Emerg. Technol. Data Mining* and Inf. Security, A. Abraham, P. Dutta, J. K. Mandal, A. Bhattacharya, and S. Dutta, Eds. Singap.: Springer Singap., 2019, pp. 859–872.
- [16] K. Merchant and Y. Pande, "Nlp based latent semantic analysis for legal text summarization," in 2018 Int. Conf. Advances in Comput., Commun. and Inform. (ICACCI), 2018, pp. 1803–1807.
- [17] D. Anand and R. Wagh, "Effective deep learning approaches for summarization of legal texts," J. of King Saud Univ. - Comput. and Inf. Sci., vol. 34, no. 5, pp. 2141–2150, 2022.
- [18] N. Alampalli Ramu, M. S. Bandarupalli, M. S. S. Nekkanti, and G. Ramesh, "Summarization of research publications using automatic extraction," in *Intell. Data Commun. Technol. and Internet of Things*, D. J. Hemanth, S. Shakya, and Z. Baig, Eds. Cham: Springer Int. Publishing, 2020, pp. 1–10.
- [19] X.-J. Jiang, X.-L. Mao, B.-S. Feng, X. Wei, B.-B. Bian, and H. Huang, "Hsds: An abstractive model for automatic survey generation," in *Database Syst. Adv. Appl.*, G. Li, J. Yang, J. Gama, J. Natwichai, and Y. Tong, Eds. Cham: Springer Int. Publishing, 2019, pp. 70–86.
- [20] Y. Liu, T. Safavi, A. Dighe, and D. Koutra, "Graph summarization methods and applications: A survey," ACM Comput. Surv., vol. 51, no. 3, jun 2018.
- [21] O. Tas and F. Kiyani, "A survey automatic text summarization," PressAcademia Procedia, vol. 5, no. 1, pp. 205–213, 2007.
- [22] S. Gholamrezazadeh, M. A. Salehi, and B. Gholamzadeh, "A comprehensive survey on text summarization systems," in 2009 2nd Int. Conf. Comput. Sci. and its Appl. IEEE, 2009, pp. 1–6.
- [23] V. Gupta and G. S. Lehal, "A survey of text summarization extractive techniques," *J. of Emerg. Technol. in Web Intell.*, vol. 2, no. 3, pp. 258–268, Aug. 2010.
- [24] Moratanch, N. and Chitrakala, S., "A survey on extractive text summarization," in 2017 Int. Conf. Comput., Commun. and Signal Process. (ICCCSP), 2017, pp. 1–6.
- [25] N. Moratanch and S. Chitrakala, "A survey on extractive text summarization," 2017 Int. Conf. Comput., Commun. and Signal Process. (ICCCSP), pp. 1–6, 2017.
- [26] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert Syst. Appl.*, vol. 165, p. 113679, 2021.
- [27] A. A. Syed, F. L. Gaol, and T. Matsuo, "A survey of the state-of-the-art models in neural abstractive text summarization," *IEEE Access*, vol. 9, pp. 13 248–13 265, 2021.
- [28] M. F. Mridha, A. A. Lima, K. Nur, S. C. Das, M. Hasan, and M. M. Kabir, "A survey of automatic text summarization: Progress, process and challenges," *IEEE Access*, vol. 9, pp. 156 043–156 070, 2021.
- [29] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," ACM Comput. Surveys, vol. 55, no. 12, pp. 1–38, 2023.
- [30] H. Y. Koh, J. Ju, M. Liu, and S. Pan, "An empirical survey on long document summarization: Datasets, models, and metrics," ACM Comput. Surv., vol. 55, no. 8, dec 2022.
- [31] H. P. Luhn, "The automatic creation of literature abstracts," IBM J. of Res. and development, vol. 2, no. 2, pp. 159–165, 1958.
- [32] A. P. Widyassari, S. Rustad, G. F. Shidik, E. Noersasongko, A. Syukur, A. Affandy et al., "Review of automatic text summarization techniques & methods," J. of King Saud Univ.-Comput. and Inf. Sci., vol. 34, no. 4, pp. 1029–1046, 2022.
- [33] L. Hou, P. Hu, and C. Bei, "Abstractive document summarization via neural model with joint attention," in *Natural Lang. Process. and Chin. Comput.: 6th CCF Int. Conf.*, NLPCC 2017, Dalian, China, November 8–12, 2017, Proc. 6. Springer, 2018, pp. 329–338.
- [34] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur,

- "Recurrent neural network based language model." in *Interspeech*, vol. 2, no. 3. Makuhari, 2010, pp. 1045–1048.
- [35] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in NIPS, 2017.
- [36] M. Kirmani, N. Manzoor Hakak, M. Mohd, and M. Mohd, "Hybrid text summarization: A survey," in *Soft Comput.: Theories and Appl.*, K. Ray, T. K. Sharma, S. Rawat, R. K. Saini, and A. Bandyopadhyay, Eds. Singap.: Springer Singap., 2019, pp. 63–73.
- [37] H. Chen, X. Liu, D. Yin, and J. Tang, "A survey on dialogue systems: Recent advances and new frontiers," SIGKDD Explor., vol. 19, no. 2, pp. 25–35, 2017.
- [38] Q. Li, Y. Chen, J. Wang, Y. Chen, and H. Chen, "Web media and stock markets: A survey and future directions from a big data perspective," *IEEE Trans. on Knowl. and Data Eng.*, vol. 30, no. 2, pp. 381–399, 2018
- [39] A. Kanapala, S. Pal, and R. Pamula, "Text summarization from legal documents: a survey," *Artif. Intell. Review*, vol. 51, no. 3, pp. 371–402, Mar 2019.
- [40] N. Ibrahim Altmami and M. El Bachir Menai, "Automatic summarization of scientific articles: A survey," J. of King Saud Univ. Comput. and Inf. Sci., vol. 34, no. 4, pp. 1011–1028, 2022.
- [41] C. Ma, W. E. Zhang, M. Guo, H. Wang, and Q. Z. Sheng, "Multi-document summarization via deep learning techniques: A survey," ACM Comput. Surv., vol. 55, no. 5, dec 2022.
- [42] R. Jain, A. Jangra, S. Saha, and A. Jatowt, "A survey on medical document summarization," 2022.
- [43] P. Bhattacharya, K. Hiware, S. Rajgaria, N. Pochhi, K. Ghosh, and S. Ghosh, "A comparative study of summarization algorithms applied to legal case judgments," in *Advances in Inf. Retrieval*, L. Azzopardi, B. Stein, N. Fuhr, P. Mayr, C. Hauff, and D. Hiemstra, Eds. Cham: Springer Int. Publishing, 2019, pp. 413–428.
- [44] D. Jain, M. D. Borah, and A. Biswas, "Summarization of legal documents: Where are we now and the way forward," *Comput. Sci. Review*, vol. 40, p. 100388, 2021.
- [45] X. Feng, X. Feng, B. Qin, and X. Geng, "Dialogue discourse-aware graph model and data augmentation for meeting summarization," in *Proc. 30th IJCAI, IJCAI-21*, Z.-H. Zhou, Ed. IJCAI Org., 8 2021, pp. 3808–3814, main Track.
- [46] X. Liu, S. Zang, C. Zhang, X. Chen, and Y. Ding, "Clts+: A new chinese long text summarization dataset with abstractive summaries," in *Int. Conf. Artif. Neural Networks*. Springer, 2022, pp. 73–84.
- [47] A. Asi, S. Wang, R. Eisenstadt, D. Geckt, Y. Kuper, Y. Mao, and R. Ronen, "An end-to-end dialogue summarization system for sales calls," in *Proc. 2022 Conf. North American Chapter Assoc. Comput. Linguistics: Human Lang. Technol.: Industry Track*, A. Loukina, R. Gangadharaiah, and B. Min, Eds. Hybrid: Seattle, Washington + Online: Assoc. Comput. Linguistics, Jul. 2022, pp. 45–53.
- [48] R. Nallapati, B. Zhou, C. dos Santos, C. Gulcehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence RNNs and beyond," in *Proc. 20th SIGNLL Conf. Computational Natural Lang. Learning*, S. Riezler and Y. Goldberg, Eds. Berlin, Germany: Assoc. Comput. Linguistics, Aug. 2016, pp. 280–290.
- [49] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics (Volume 1: Long Papers)*, R. Barzilay and M.-Y. Kan, Eds. Vancouver, Canada: Assoc. Comput. Linguistics, Jul. 2017, pp. 1073–1083.
- [50] S. Narayan, J. Maynez, J. Adamek, D. Pighin, B. Bratanic, and R. McDonald, "Stepwise extractive summarization and planning with structured transformers," in *EMNLP*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Online: Assoc. Comput. Linguistics, Nov. 2020, pp. 4143–4159.
- [51] M. Zhong, P. Liu, Y. Chen, D. Wang, X. Qiu, and X. Huang, "Extractive summarization as text matching," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Assoc. Comput. Linguistics, Jul. 2020, pp. 6197–6208.
- [52] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," in *EMNLP*, L. Marquez, C. Callison-Burch, and J. Su, Eds. Lisbon, Portugal: Assoc. Comput. Linguistics, Sep. 2015, pp. 379–389.
- [53] U. Khandelwal, K. Clark, D. Jurafsky, and L. Kaiser, "Sample efficient text summarization using a single pre-trained transformer," arXiv preprint arXiv:1905.08836, 2019.
- [54] P. Li, W. Lam, L. Bing, and Z. Wang, "Deep recurrent generative decoder for abstractive text summarization," in *EMNLP*, M. Palmer, R. Hwa, and S. Riedel, Eds. Copenhagen, Denmark: Assoc. Comput.

- Linguistics, Sep. 2017, pp. 2091-2100.
- [55] D. Graff, J. Kong, K. Chen, and K. Maeda, "English gigaword," Linguistic Data Consortium, Philadelphia, vol. 4, no. 1, p. 34, 2003.
- [56] K. Song, B. Wang, Z. Feng, and F. Liu, "A new approach to overgenerating and scoring abstractive summaries," in *Proc. 2021 Conf. North American Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, Eds. Online: Assoc. Comput. Linguistics, Jun. 2021, pp. 1392–1404.
- [57] K. Song, B. Wang, Z. Feng, R. Liu, and F. Liu, "Controlling the amount of verbatim copying in abstractive summarization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 05, 2020, pp. 8902–8909.
- [58] S. Narayan, S. B. Cohen, and M. Lapata, "Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization," in *EMNLP*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds. Brussels, Belgium: Assoc. Comput. Linguistics, Oct.-Nov. 2018, pp. 1797–1807.
- [59] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," in *EMNLP-IJCNLP*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Assoc. Comput. Linguistics, Nov. 2019, pp. 3730–3740.
- [60] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "Pegasus: pre-training with extracted gap-sentences for abstractive summarization," in *Proc. 37th Int. Conf. Machine Learning*, ser. ICML'20. JMLR.org, 2020.
- [61] A. Fabbri, I. Li, T. She, S. Li, and D. Radev, "Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds. Florence, Italy: Assoc. Comput. Linguistics, Jul. 2019, pp. 1074–1084.
- [62] P. He, B. Peng, L. Lu, S. Wang, J. Mei, Y. Liu, R. Xu, H. H. Awadalla, Y. Shi, C. Zhu et al., "Z-code++: A pre-trained language model optimized for abstractive summarization," arXiv preprint arXiv:2208.09770, 2022.
- [63] D. Wang, P. Liu, Y. Zheng, X. Qiu, and X. Huang, "Heterogeneous graph neural networks for extractive document summarization," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Assoc. Comput. Linguistics, Jul. 2020, pp. 6209–6219.
- [64] M. Yasunaga, J. Kasai, R. Zhang, A. R. Fabbri, I. Li, D. Friedman, and D. R. Radev, "Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 01, pp. 7386–7393, Jul. 2019.
- [65] I. Cachola, K. Lo, A. Cohan, and D. Weld, "TLDR: Extreme summarization of scientific documents," in *Findings Assoc. Comput. Linguistics: EMNLP*, T. Cohn, Y. He, and Y. Liu, Eds. Online: Assoc. Comput. Linguistics, Nov. 2020, pp. 4766–4777.
- [66] A. Ni, Z. Azerbayev, M. Mutuma, T. Feng, Y. Zhang, T. Yu, A. H. Awadallah, and D. Radev, "SummerTime: Text summarization toolkit for non-experts," in *EMNLP*, H. Adel and S. Shi, Eds. Online and Punta Cana, Dominican Republic: Assoc. Comput. Linguistics, Nov. 2021, pp. 329–338.
- [67] A. Cohan, F. Dernoncourt, D. S. Kim, T. Bui, S. Kim, W. Chang, and N. Goharian, "A discourse-aware attention model for abstractive summarization of long documents," in *Proc. 2018 Conf. North American Chapter Assoc. Comput. Linguistics: Human Lang. Technol., Volume 2* (*Short Papers*), M. Walker, H. Ji, and A. Stent, Eds. New Orleans, Louisiana: Assoc. Comput. Linguistics, Jun. 2018, pp. 615–621.
- [68] W. Xiao, I. Beltagy, G. Carenini, and A. Cohan, "PRIMERA: Pyramid-based masked sentence pre-training for multi-document summarization," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Assoc. Comput. Linguistics, May 2022, pp. 5245–5263
- [69] S. Sotudeh, A. Cohan, and N. Goharian, "On generating extended summaries of long documents," arXiv preprint arXiv:2012.14136, 2020
- [70] M. Koupaee and W. Y. Wang, "Wikihow: A large scale text summarization dataset," ArXiv, vol. abs/1810.09305, 2018.
- [71] A. Savelieva, B. Au-Yeung, and V. Ramani, "Abstractive summarization of spoken and written instructions with bert," arXiv preprint arXiv:2008.09676, 2020.
- [72] J. Lin, X. Sun, S. Ma, and Q. Su, "Global encoding for abstractive summarization," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics (Volume 2: Short Papers)*, I. Gurevych and Y. Miyao, Eds. Melbourne, Australia: Assoc. Comput. Linguistics, Jul. 2018, pp. 163–169.
- [73] S. Ma, X. Sun, J. Lin, and H. Wang, "Autoencoder as assistant supervisor: Improving text representation for Chinese social media

- text summarization," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics (Volume 2: Short Papers)*, I. Gurevych and Y. Miyao, Eds. Melbourne, Australia: Assoc. Comput. Linguistics, Jul. 2018, pp. 725–731.
- [74] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," *Advances in neural Inf. Process. systems*, vol. 28, 2015.
- [75] D. Harman and P. Over, "The effects of human variation in duc summarization evaluation," in *Text Summarization Branches Out*, 2004, pp. 10–17.
- [76] B. Hu, Q. Chen, and F. Zhu, "LCSTS: A large scale Chinese short text summarization dataset," in *EMNLP*, L. Màrquez, C. Callison-Burch, and J. Su, Eds. Lisbon, Portugal: Assoc. Comput. Linguistics, Sep. 2015, pp. 1967–1972.
- [77] X. Feng, X. Feng, L. Qin, B. Qin, and T. Liu, "Language model as an annotator: Exploring DialoGPT for dialogue summarization," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics and the 11th Int. Joint Conf. Natural Lang. Process. (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Online: Assoc. Comput. Linguistics, Aug. 2021, pp. 1479–1491.
- [78] Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan, "DIALOGPT: Large-scale generative pre-training for conversational response generation," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics: System Demonstrations*, A. Celikyilmaz and T.-H. Wen, Eds. Online: Assoc. Comput. Linguistics, Jul. 2020, pp. 270–278.
- [79] B. Chintagunta, N. Katariya, X. Amatriain, and A. Kannan, "Medically aware GPT-3 as a data generator for medical dialogue summarization," in *Proc. 2nd Workshop on Natural Lang. Process. for Medical Conversations*, C. Shivade, R. Gangadharaiah, S. Gella, S. Konam, S. Yuan, Y. Zhang, P. Bhatia, and B. Wallace, Eds. Online: Assoc. Comput. Linguistics, Jun. 2021, pp. 66–76.
- [80] A. Liu, S. Swayamdipta, N. A. Smith, and Y. Choi, "Wanli: Worker and ai collaboration for natural language inference dataset creation," arXiv preprint arXiv:2201.05955, 2022.
- [81] T. Goyal and G. Durrett, "Annotating and modeling fine-grained factuality in summarization," in *Proc. 2021 Conf. North American Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, Eds. Online: Assoc. Comput. Linguistics, Jun. 2021, pp. 1449–1462.
- [82] W. Kryscinski, B. McCann, C. Xiong, and R. Socher, "Evaluating the factual consistency of abstractive text summarization," in *EMNLP*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Online: Assoc. Comput. Linguistics, Nov. 2020, pp. 9332–9346.
- [83] T. Goyal and G. Durrett, "Evaluating factuality in generation with dependency-level entailment," in *Findings Assoc. Comput. Linguistics:* EMNLP, 2020, pp. 3592–3603.
- [84] V. Balachandran, H. Hajishirzi, W. Cohen, and Y. Tsvetkov, "Correcting diverse factual errors in abstractive summarization via post-editing and language model infilling," in *EMNLP*, 2022, pp. 9818–9830.
- [85] T. Vodolazova, E. Lloret, R. Muñoz, and M. Palomar, "The role of statistical and semantic features in single-document extractive summarization," *Artif. Intell. Res.*, vol. 2, pp. 35–44, 2013.
- [86] H. Saif, M. Fernández, Y. He, and H. Alani, "On stopwords, filtering and data sparsity for sentiment analysis of twitter," in *Int. Conf. Lang. Resources and Evaluation*, 2014.
- [87] E. Charniak, "Statistical techniques for natural language parsing," AI Mag., vol. 18, no. 4, pp. 33–44, 1997.
- [88] M. Y. Nuzumlali and A. Özgür, "Analyzing stemming approaches for turkish multi-document summarization," in *Conf. EMNLP*, 2014.
- [89] E. Galiotou, N. N. Karanikolas, and C. Tsoulloftas, "On the effect of stemming algorithms on extractive summarization: a case study," in *Panhellenic Conf. Inform.*, 2013.
- [90] L. Hickman, S. Thapa, L. Tay, M. Cao, and P. Srinivasan, "Text preprocessing for text mining in organizational research: Review and recommendations," *Organizational Res. Methods*, vol. 25, no. 1, pp. 114–146, 2022.
- [91] J.-M. Torres-Moreno, "Beyond stemming and lemmatization: Ultrastemming to improve automatic text summarization," ArXiv, vol. abs/1209.3126, 2012.
- [92] V. K. Gupta and T. J. Siddiqui, "Multi-document summarization using sentence clustering," in 2012 4th Int. Conf. Intell. Human Comput. Interaction (IHCI). IEEE, 2012, pp. 1–5.
- [93] G. Moro and L. Ragazzi, "Semantic self-segmentation for abstractive summarization of long documents in low-resource regimes," in AAAI Conf. Artif. Intell., 2022.

- [94] J. Wang, J. Tan, H. Jin, and S. Qi, "Unsupervised graph-clustering learning framework for financial news summarization," 2021 Int. Conf. Data Mining Workshops (ICDMW), pp. 719–726, 2021.
- [95] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics (Volume 1: Long Papers)*, K. Erk and N. A. Smith, Eds. Berlin, Germany: Assoc. Comput. Linguistics, Aug. 2016, pp. 1715–1725.
- [96] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. R. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. S. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," ArXiv, vol. abs/1609.08144, 2016.
- [97] S. Bird, E. Klein, and E. Loper, Natural language processing with Python: analyzing text with the natural language toolkit. "O'Reilly Media, Inc.", 2009.
- [98] H. He and J. D. Choi, "The stem cell hypothesis: Dilemma behind multi-task learning with transformer encoders," in EMNLP. Online and Punta Cana, Dominican Republic: Assoc. Comput. Linguistics, Nov. 2021, pp. 5555–5577.
- [99] B. Gong, W.-L. Chao, K. Grauman, and F. Sha, "Diverse sequential subset selection for supervised video summarization," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.- Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, p. 2069–2077.
- [100] K. Gunaratna, K. Thirunarayan, and A. Sheth, "Faces: Diversity-aware entity summarization using incremental hierarchical conceptual clustering," *Proc. AAAI Conf. Artif. Intell.*, vol. 29, no. 1, Feb. 2015.
- [101] M. Peyrard, "A simple theoretical model of importance for summarization," in *Proceedings 57th Annu. Meeting Assoc. Comput. Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds. Florence, Italy: Assoc. Comput. Linguistics, Jul. 2019, pp. 1059–1073.
- [102] R. Yan, H. Jiang, M. Lapata, S.-D. Lin, X. Lv, and X. Li, "i, poet: automatic chinese poetry composition through a generative summarization framework under constrained optimization," in *Proc. 23rd IJCAI*, ser. IJCAI '13. AAAI Press, 2013, p. 2197–2203.
- [103] K. Wang, T. Liu, Z. Sui, and B. Chang, "Affinity-preserving random walk for multi-document summarization," in *EMNLP*, M. Palmer, R. Hwa, and S. Riedel, Eds. Copenhagen, Denmark: Assoc. Comput. Linguistics, Sep. 2017, pp. 210–220.
- [104] G. Shang, W. Ding, Z. Zhang, A. Tixier, P. Meladianos, M. Vazir-giannis, and J.-P. Lorré, "Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics (Volume 1: Long Papers)*, I. Gurevych and Y. Miyao, Eds. Melbourne, Australia: Assoc. Comput. Linguistics, Jul. 2018, pp. 664–674.
- [105] H. Christian, M. P. Agus, and D. Suhartono, "Single document automatic text summarization using term frequency-inverse document frequency (tf-idf)," *ComTech*, vol. 7, no. 4, 2016.
- [106] N. Alsaedi, P. Burnap, and O. Rana, "Temporal tf-idf: A high performance approach for event summarization in twitter," in 2016 IEEE/WIC/ACM Int. Conf. Web Intell. (WI), 2016, pp. 515–521.
- [107] C. Rioux, S. A. Hasan, and Y. Chali, "Fear the REAPER: A system for automatic multi-document summarization with reinforcement learning," in *EMNLP*, A. Moschitti, B. Pang, and W. Daelemans, Eds. Doha, Qatar: Assoc. Comput. Linguistics, Oct. 2014, pp. 681–690.
- [108] W. Luo, F. Liu, Z. Liu, and D. Litman, "Automatic summarization of student course feedback," in *Proc. 2016 Conf. North American Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, K. Knight, A. Nenkova, and O. Rambow, Eds. San Diego, California: Assoc. Comput. Linguistics, Jun. 2016, pp. 80–85.
- [109] X. Qian and Y. Liu, "Fast joint compression and summarization via graph cuts," in *EMNLP*, D. Yarowsky, T. Baldwin, A. Korhonen, K. Livescu, and S. Bethard, Eds. Seattle, Washington, USA: Assoc. Comput. Linguistics, Oct. 2013, pp. 1492–1502.
- [110] C.-Y. Lin and E. Hovy, "Automatic evaluation of summaries using n-gram co-occurrence statistics," in *Proc. 2003 Human Lang. Technol. Conf. North American Chapter Assoc. Comput. Linguistics*, 2003, pp. 150–157.
- [111] M. Banko and L. Vanderwende, "Using n-grams to understand the nature of summaries," in *Proc. of HLT-NAACL 2004: Short Papers*, ser. HLT-NAACL-Short '04. USA: Assoc. Comput. Linguistics, 2004, p. 1–4.
- [112] T. K. Landauer and S. T. Dumais, "Latent semantic analysis," Scholarpedia, vol. 3, p. 4356, 2008.

- [113] J.-Y. Yeh, H.-R. Ke, W.-P. Yang, and I.-H. Meng, "Text summarization using a trainable summarizer and latent semantic analysis," *Inf. Process.* & *Manage.*, vol. 41, no. 1, pp. 75–95, 2005, an Asian Digital Libraries Perspective.
- [114] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," J. of machine Learning Res., vol. 3, no. Jan, pp. 993–1022, 2003.
- [115] A. Haghighi and L. Vanderwende, "Exploring content models for multi-document summarization," in *Proc. of human Lang. Technol.: The 2009 Annu. Conf. North American Chapter Assoc. Comput. Linguistics*, 2009, pp. 362–370.
  [116] Y. Lu, C. Zhai, and N. Sundaresan, "Rated aspect summarization of
- [116] Y. Lu, C. Zhai, and N. Sundaresan, "Rated aspect summarization of short comments," in *Proc. 18th Int. Conf. World Wide Web*, ser. WWW '09. New York, NY, USA: Assoc. for Comput. Machinery, 2009, p. 131–140.
- [117] C. Allen and T. M. Hospedales, "Analogies explained: Towards understanding word embeddings," in *Int. Conf. Machine Learning*, 2019.
- [118] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Int. Conf. Learning Representations*, 2013.
- [119] G. Rossiello, P. Basile, and G. Semeraro, "Centroid-based text summarization through compositionality of word embeddings," in *Proc. MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, G. Giannakopoulos, E. Lloret, J. M. Conroy, J. Steinberger, M. Litvak, P. Rankel, and B. Favre, Eds. Valencia, Spain: Assoc. Comput. Linguistics, Apr. 2017, pp. 12–21.
- [120] M. M. Haider, M. A. Hossin, H. R. Mahi, and H. Arif, "Automatic text summarization using gensim word2vec and k-means clustering algorithm," in 2020 IEEE Region 10 Symposium (TENSYMP). IEEE, 2020, pp. 283–286.
- [121] S. Ji, N. Satish, S. Li, and P. K. Dubey, "Parallelizing word2vec in shared and distributed memory," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 9, p. 2090–2100, sep 2019.
- [122] U. Naseem, I. Razzak, S. K. Khan, and M. Prasad, "A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models," ACM Trans. Asian Low-Resour. Lang. Inf. Process., vol. 20, no. 5, jun 2021.
- [123] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in EMNLP, A. Moschitti, B. Pang, and W. Daelemans, Eds. Doha, Qatar: Assoc. Comput. Linguistics, Oct. 2014, pp. 1532–1543.
- [124] W. Xiao and G. Carenini, "Extractive summarization of long documents by combining global and local context," in *EMNLP-IJCNLP*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Assoc. Comput. Linguistics, Nov. 2019, pp. 3011–3021.
- [125] S. S. Nath and B. Roy, "Towards automatically generating release notes using extractive summarization technique," arXiv preprint arXiv:2204.05345, 2022.
- [126] Y. Wang, Y. Hou, W. Che, and T. Liu, "From static to dynamic word representations: a survey," *Int. J. Mach. Learn. Cybern.*, vol. 11, no. 7, pp. 1611–1630, 2020.
- [127] H. Zheng and M. Lapata, "Sentence centrality revisited for unsupervised summarization," in *Proc.57th Annu. Meeting Assoc. Comput. Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds. Florence, Italy: Assoc. Comput. Linguistics, Jul. 2019, pp. 6236–6247.
- [128] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pretraining of deep bidirectional transformers for language understanding," in *Proc. 2019 Conf. North American Chapter Assoc. Comput. Linguistics: Human Lang. Technol., Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Assoc. Comput. Linguistics, Jun. 2019, pp. 4171–4186.
- [129] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in Advances in Neural Inf. Process. Systems, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901.
- [130] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," *Science China Technol. Sci.*, vol. 63, no. 10, pp. 1872–1897, 2020.
- [131] V. Joshi, M. Peters, and M. Hopkins, "Extending a parser to distant domains using a few dozen partially annotated examples," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics (Volume 1: Long Papers)*, I. Gurevych and Y. Miyao, Eds. Melbourne, Australia: Assoc. Comput. Linguistics, Jul. 2018, pp. 1190–1199.

- [132] J. Zhou and A. M. Rush, "Simple unsupervised summarization by contextual matching," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 5101–5106.
- [133] K. Ethayarajh, "How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings," in *EMNLP-IJCNLP*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Assoc. Comput. Linguistics, Nov. 2019, pp. 55–65.
- [134] X. Zhang, F. Wei, and M. Zhou, "HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization," in *Proceedings 57th Annu. Meeting Assoc. Comput. Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds. Florence, Italy: Assoc. Comput. Linguistics, Jul. 2019, pp. 5059–5069.
- [135] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," *ArXiv*, vol. abs/2302.13971, 2023.
- [136] Y. Dong, J.-B. Cordonnier, and A. Loukas, "Attention is not all you need: Pure attention loses rank doubly exponentially with depth," in *Int. Conf. Machine Learning*. PMLR, 2021, pp. 2793–2803.
- [137] T. R. Goodwin, M. E. Savery, and D. Demner-Fushman, "Flight of the pegasus? comparing transformers on few-shot and zero-shot multidocument abstractive summarization," *Proc. of COLING. Int. Conf. Comput. Linguistics*, vol. 2020, pp. 5640 – 5646, 2020.
- [138] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Assoc. Comput. Linguistics, Jul. 2020, pp. 7871–7880.
- [139] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The J. of Machine Learning Res.*, vol. 21, no. 1, pp. 5485–5551, 2020.
- [140] X. Yang, Y. Li, X. Zhang, H. Chen, and W. Cheng, "Exploring the limits of chatgpt for query or aspect-based text summarization," *ArXiv*, vol. abs/2302.08081, 2023.
- [141] R. Mihalcea and P. Tarau, "TextRank: Bringing order into text," in EMNLP, D. Lin and D. Wu, Eds. Barcelona, Spain: Assoc. Comput. Linguistics, Jul. 2004, pp. 404–411.
- [142] G. Erkan and D. R. Radev, "Lexrank: graph-based lexical centrality as salience in text summarization," *J. Artif. Int. Res.*, vol. 22, no. 1, p. 457–479, dec 2004.
- [143] D. Gillick and B. Favre, "A scalable global model for summarization," in *Proc. Workshop on Integer Linear Programming for Natural Lang. Process.*, J. Clarke and S. Riedel, Eds. Boulder, Colorado: Assoc. Comput. Linguistics, Jun. 2009, pp. 10–18.
- [144] Y. Zhang, Y. Xia, Y. Liu, and W. Wang, "Clustering sentences with density peaks for multi-document summarization," in *Proc. 2015 Conf. North American Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, R. Mihalcea, J. Chai, and A. Sarkar, Eds. Denver, Colorado: Assoc. Comput. Linguistics, May–Jun. 2015, pp. 1262–1267.
- [145] S. M. Mohammed, K. Jacksi, and S. R. Zeebaree, "Glove word embedding and dbscan algorithms for semantic document clustering," in 2020 Int. Conf. Adv. Sci. and Eng. (ICOASE). IEEE, 2020, pp. 1–6.
- [146] S. Abdel-Salam and A. Rafea, "Performance study on extractive text summarization using bert models," *Inf.*, vol. 13, no. 2, p. 67, 2022.
- [147] Q. Xie, J. A. Bishop, P. Tiwari, and S. Ananiadou, "Pre-trained language models with domain knowledge for biomedical extractive summarization," *Knowl.-Based Systems*, vol. 252, p. 109460, 2022.
- [148] X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W.-K. Wong, and W. chun Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in NIPS, 2015.
- [149] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in NIPS 2014 Workshop on Deep Learning, December 2014, 2014.
- [150] R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: a recurrent neural network based sequence model for extractive summarization of documents," in *Proc. Thirty-1st AAAI Conf. Artif. Intell.*, ser. AAAI'17. AAAI Press, 2017, p. 3075–3081.
- [151] S. Chopra, M. Auli, and A. M. Rush, "Abstractive sentence summarization with attentive recurrent neural networks," in *Proc. 2016 Conf. North American Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, K. Knight, A. Nenkova, and O. Rambow, Eds. San Diego, California: Assoc. Comput. Linguistics, Jun. 2016, pp. 93–98.
- [152] B. Lindemann, B. Maschler, N. Sahlab, and M. Weyrich, "A survey on anomaly detection for technical syst.using lstm networks," *Comput. Ind.*, vol. 131, p. 103498, 2021.

- [153] P. M. Hanunggul and S. Suyanto, "The impact of local attention in lstm for abstractive text summarization," 2019 Int. Seminar on Res. of Inf. Technol. and Intell. Syst. (ISRITI), pp. 54–57, 2019.
- [154] Y. Zhang, J. Liao, J. Tang, W. D. Xiao, and Y. Wang, "Extractive document summarization based on hierarchical gru," 2018 Int. Conf. Robots & Intell. Syst. (ICRIS), pp. 341–346, 2018.
- [155] Q. Grail, J. Perez, and E. Gaussier, "Globalizing bert-based transformer architectures for long document summarization," in *Proc. 16th Conf. European chapter Assoc. Comput. Linguistics: Main volume*, 2021, pp. 1792–1810.
- [156] B. Pang, E. Nijkamp, W. Kryscinski, S. Savarese, Y. Zhou, and C. Xiong, "Long document summarization with top-down and bottomup inference," in *Findings Assoc. Comput. Linguistics: EACL 2023*, A. Vlachos and I. Augenstein, Eds. Dubrovnik, Croatia: Assoc. Comput. Linguistics, May 2023, pp. 1267–1284.
- [157] Z. Wang, Z. Duan, H. Zhang, C. Wang, L. Tian, B. Chen, and M. Zhou, "Friendly topic assistant for transformer based abstractive summarization," in *EMNLP*, 2020, pp. 485–497.
- [158] A. Pagnoni, A. R. Fabbri, W. Kryściński, and C.-S. Wu, "Socratic pretraining: Question-driven pretraining for controllable summarization," arXiv preprint arXiv:2212.10449, 2022.
- [159] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J. Nie, and J. rong Wen, "A survey of large language models," *ArXiv*, vol. abs/2303.18223, 2023.
- [160] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *Int. Conf. Machine Learning*. PMLR, 2019, pp. 2790–2799.
- [161] L. Tang, Z. Sun, B. Idnay, J. G. Nestor, A. Soroush, P. A. Elias, Z. Xu, Y. Ding, G. Durrett, J. F. Rousseau, C. Weng, and Y. Peng, "Evaluating large language models on medical evidence summarization," npj Digital Medicine, vol. 6, no. 1, p. 158, 2023.
- [162] M. T. R. Laskar, M. S. Bari, M. Rahman, M. A. H. Bhuiyan, S. Joty, and J. X. Huang, "A systematic study and comprehensive evaluation of chatgpt on benchmark datasets," arXiv preprint arXiv:2305.18486, 2023
- [163] T. Zhang, F. Ladhak, E. Durmus, P. Liang, K. McKeown, and T. B. Hashimoto, "Benchmarking large language models for news summarization," 2023.
- [164] M. Ravaut, S. Joty, A. Sun, and N. F. Chen, "On context utilization in summarization with large language models," 2023.
- [165] L. Basyal and M. Sanghvi, "Text summarization using large language models: A comparative study of mpt-7b-instruct, falcon-7b-instruct, and openai chat-gpt models," 2023.
- [166] T. Goyal, J. J. Li, and G. Durrett, "News summarization and evaluation in the era of gpt-3," 2023.
- [167] N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano, "Learning to summarize with human feedback," *Advances in Neural Inf. Process. Systems*, vol. 33, pp. 3008–3021, 2020.
- [168] J. Wu, L. Ouyang, D. M. Ziegler, N. Stiennon, R. Lowe, J. Leike, and P. Christiano, "Recursively summarizing books with human feedback," 2021
- [169] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *Int. J. Comput. Vision*, vol. 129, pp. 1789–1819, 2021.
- [170] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," stat, vol. 1050, p. 9, 2015.
- [171] M. Sclar, P. West, S. Kumar, Y. Tsvetkov, and Y. Choi, "Referee: Reference-free sentence summarization with sharper controllability through symbolic knowledge distillation," in *EMNLP*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Assoc. Comput. Linguistics, Dec. 2022, pp. 9649–9668.
- [172] Y. Xu, R. Xu, D. Iter, Y. Liu, S. Wang, C. Zhu, and M. Zeng, "Inheritsumm: A general, versatile and compact summarizer by distilling from gpt," 2023.
- [173] A. Brazinskas, R. Nallapati, M. Bansal, and M. Dreyer, "Efficient few-shot fine-tuning for opinion summarization," in *Findings Assoc. Comput. Linguistics: NAACL 2022*, M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, Eds. Seattle, United States: Assoc. Comput. Linguistics, Jul. 2022, pp. 1509–1523.
- [174] D. F. Navarro, M. Dras, and S. Berkovsky, "Few-shot fine-tuning SOTA summarization models for medical dialogues," in *Proc. 2022 Conf. North American Chapter Assoc. Comput. Linguistics: Human Lang. Technol.: Student Res. Workshop*, D. Ippolito, L. H. Li, M. L. Pacheco, D. Chen, and N. Xue, Eds. Hybrid: Seattle, Washington + Online:

- Assoc. Comput. Linguistics, Jul. 2022, pp. 254–266.
- [175] Y. Zhang, X. Zhang, X. Wang, S.-q. Chen, and F. Wei, "Latent prompt tuning for text summarization," arXiv preprint arXiv:2211.01837, 2022.
- [176] Y. Chen, Y. Liu, R. Xu, Z. Yang, C. Zhu, M. Zeng, and Y. Zhang, "Unisumm: Unified few-shot summarization with multi-task pretraining and prefix-tuning," arXiv preprint arXiv:2211.09783, 2022.
- [177] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pretrain, prompt, and predict: A systematic survey of prompting methods in natural language processing," ACM Comput. Surv., vol. 55, no. 9, ian 2023.
- [178] S. Narayan, Y. Zhao, J. Maynez, G. Simões, V. Nikolaev, and R. McDonald, "Planning with learned entity prompts for abstractive summarization," *Trans. Assoc. Comput. Linguistics*, vol. 9, pp. 1475– 1492, 2021.
- [179] R. Shin, C. Lin, S. Thomson, C. Chen, S. Roy, E. A. Platanios, A. Pauls, D. Klein, J. Eisner, and B. Van Durme, "Constrained language models yield few-shot semantic parsers," in *EMNLP*. Assoc. Comput. Linguistics, 2021.
- [180] Z. Jiang, F. F. Xu, J. Araki, and G. Neubig, "How can we know what language models know?" *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 423–438, 2020.
- [181] Y. Zhou, K. Shi, W. Zhang, Y. Liu, Y. Zhao, and A. Cohan, "Odsum: New benchmarks for open domain multi-document summarization," 2023
- [182] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou et al., "Chain-of-thought prompting elicits reasoning in large language models," Advances in Neural Inf. Process. Systems, vol. 35, pp. 24 824–24 837, 2022.
- [183] Y. Wang, Z. Zhang, and R. Wang. Element-aware summarization with large language models: Expert-aligned evaluation and chain-of-thought method.
- [184] G. Adams, A. Fabbri, F. Ladhak, E. Lehman, and N. Elhadad, "From sparse to dense: GPT-4 summarization with chain of density prompting," in *Proc. 4th New Frontiers in Summarization Workshop*, Y. Dong, W. Xiao, L. Wang, F. Liu, and G. Carenini, Eds. Singap.: Assoc. Comput. Linguistics, Dec. 2023, pp. 68–74.
- [185] Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou, R. Zheng, X. Fan, X. Wang, L. Xiong, Y. Zhou, W. Wang, C. Jiang, Y. Zou, X. Liu, Z. Yin, S. Dou, R. Weng, W. Cheng, Q. Zhang, W. Qin, Y. Zheng, X. Qiu, X. Huang, and T. Gui, "The rise and potential of large language model based agents: A survey," 2023.
- [186] W. Xiao, Y. Xie, G. Carenini, and P. He, "Chatgpt-steered editing instructor for customization of abstractive summarization," 2023.
- [187] X. Pu, M. Gao, and X. Wan, "Summarization is (almost) dead," 2023.
- [188] L. Luo, Y.-F. Li, G. Haffari, and S. Pan, "Reasoning on graphs: Faithful and interpretable large language model reasoning," *arXiv preprint arXiv:2310.01061*, 2023.
- [189] L. Ermakova, J. V. Cossu, and J. Mothe, "A survey on evaluation of summarization methods," *Inf. Process. & Manage.*, vol. 56, no. 5, pp. 1794–1814, 2019.
- [190] S. Gehrmann, Y. Deng, and A. Rush, "Bottom-up abstractive summarization," in *EMNLP*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds. Brussels, Belgium: Assoc. Comput. Linguistics, Oct.-Nov. 2018, pp. 4098–4109.
- [191] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.
- [192] C.-Y. Lin and F. Och, "Looking for a few good metrics: Rouge and its evaluation," in *Ntcir workshop*, 2004.
- [193] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting on Assoc. Comput. Linguistics - ACL '02*. Assoc. Comput. Linguistics, 2001, p. 311.
- [194] G. Doddington, "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics," in *Proc. 2nd Int. Conf. Human Lang. Technol. Research*, 2002, pp. 138–145.
- [195] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proc. acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72.
- [196] T. Zhang\*, V. Kishore\*, F. Wu\*, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," in *Int. Conf. Learning Representations*, 2020.
- [197] S. Sun and A. Nenkova, "The feasibility of embedding based automatic evaluation for single document summarization," in *EMNLP-IJCNLP*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Assoc. Comput. Linguistics, Nov. 2019, pp. 1216–1221.
- [198] M. Gao, J. Ruan, R. Sun, X. Yin, S. Yang, and X. Wan, "Human-like

- summarization evaluation with chatgpt," 2023.
- [199] S. Jain, V. Keshava, S. M. Sathyendra, P. Fernandes, P. Liu, G. Neubig, and C. Zhou, "Multi-dimensional evaluation of text summarization with in-context learning," 2023.
- [200] N. Wu, M. Gong, L. Shou, S. Liang, and D. Jiang, "Large language models are diverse role-players for summarization evaluation," 2023.
- [201] Y. Chang, K. Lo, T. Goyal, and M. Iyyer, "Booookscore: A systematic exploration of book-length summarization in the era of llms," 2023.
- [202] Z. Luo, Q. Xie, and S. Ananiadou, "Chatgpt as a factual inconsistency evaluator for text summarization," 2023.
- [203] D. Tam, A. Mascarenhas, S. Zhang, S. Kwan, M. Bansal, and C. Raffel, "Evaluating the factual consistency of large language models through news summarization," in *Findings Assoc. Comput. Linguistics: ACL* 2023, 2023, pp. 5220–5255.
- [204] Z. Gekhman, J. Herzig, R. Aharoni, C. Elkind, and I. Szpektor. Trueteacher: Learning factual consistency evaluation with large language models.
- [205] Q. Jia, S. Ren, Y. Liu, and K. Q. Zhu, "Zero-shot faithfulness evaluation for text summarization with foundation language model," in *The 2023 Conf. EMNLP*, 2023.
- [206] R. Barzilay and K. R. McKeown, "Sentence fusion for multidocument news summarization," *Computational Linguistics*, vol. 31, no. 3, pp. 297–328, 2005.
- [207] K. R. McKeown, R. Barzilay, D. Evans, V. Hatzivassiloglou, J. L. Klavans, A. Nenkova, C. Sable, B. Schiffman, and S. Sigelman, "Tracking and summarizing news on a daily basis with columbia's newsblaster," in *Proc. 2nd Int. Conf. Human Lang. Technol. Res.*, ser. HLT '02. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, p. 280–285.
- [208] D. Gholipour Ghalandari and G. Ifrim, "Examining the state-of-the-art in news timeline summarization," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Assoc. Comput. Linguistics, Jul. 2020, pp. 1322–1334.
- [209] F. Ladhak, B. Li, Y. Al-Onaizan, and K. McKeown, "Exploring content selection in summarization of novel chapters," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Assoc. Comput. Linguistics, Jul. 2020, pp. 5043–5054.
- [210] C. An, M. Zhong, Y. Chen, D. Wang, X. Qiu, and X. Huang, "Enhancing scientific papers summarization with citation graph," in Proc. AAAI Conf. Artif. Intell., vol. 35, no. 14, 2021, pp. 12498–12506.
- [211] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Int. Conf. Learning Representations*, 2018.
- [212] S. Qi, L. Li, Y. Li, J. Jiang, D. Hu, Y. Li, Y. Zhu, Y. Zhou, M. Litvak, and N. Vanetik, "Sapgraph: Structure-aware extractive summarization for scientific papers with heterogeneous graph," in *Proc. 2nd Conf. Asia-Pacific Chapter Assoc. Comput. Linguistics and the 12th Int. Joint Conf. Natural Lang. Process.*, 2022, pp. 575–586.
- [213] C. Shen, F. Liu, F. Weng, and T. Li, "A participant-based approach for event summarization using twitter streams," in *Proc. 2013 Conf. North American Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, 2013, pp. 1152–1162.
- [214] D. Inouye and J. K. Kalita, "Comparing twitter summarization algorithms for multiple post summaries," in 2011 IEEE 3rd Int. Conf. privacy, security, risk and trust and 2011 IEEE 3rd Int. Conf. social Comput. IEEE, 2011, pp. 298–306.
- [215] L. P. Kumar and A. Kabiri, "Meeting summarization: A survey of the state of the art," 2022.
- [216] Y. Mehdad, G. Carenini, and R. T. Ng, "Abstractive summarization of spoken and written conversations based on phrasal queries," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics (Volume 1: Long Papers)*, K. Toutanova and H. Wu, Eds. Baltimore, Maryland: Assoc. Comput. Linguistics, Jun. 2014, pp. 1220–1230.
- [217] P. Ganesh and S. Dingliwal. Restructuring conversations using discourse relations for zero-shot abstractive dialogue summarization.
- [218] C. E. Kahn, C. P. Langlotz, E. S. Burnside, J. A. Carrino, D. S. Channin, D. M. Hovsepian, and D. L. Rubin, "Toward best practices in radiology reporting," *Radiology*, vol. 252, no. 3, pp. 852–856, 2009.
- [219] S. Sotudeh Gharebagh, N. Goharian, and R. Filice, "Attend to medical ontologies: Content selection for clinical abstractive summarization," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Assoc. Comput. Linguistics, Jul. 2020, pp. 1899–1905.
- [220] M. Adler, J. Berant, and I. Dagan, "Entailment-based text exploration with application to the health-care domain," in *Proc. ACL 2012 System Demonstrations*, M. Zhang, Ed. Jeju Island, Korea: Assoc. Comput.

- Linguistics, Jul. 2012, pp. 79-84.
- [221] Y. Zhang, D. Merck, E. Tsai, C. D. Manning, and C. Langlotz, "Optimizing the factual correctness of a summary: A study of summarizing radiology reports," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Assoc. Comput. Linguistics, Jul. 2020, pp. 5108–5120.

# X. Biography Section



Hanlei Jin is currently working toward the Ph.D. in Management Science and Engineering at Southwestern University of Finance and Economics, Chengdu, China. He holds a B.S. degree in Management from the same academic institution. His research interests include Text Mining, Text Generation, and Financial Intelligence.



Yang Zhang received his Ph.D. from the Graduate School of Informatics, Kyoto University in 2022. Before that, he received Bachelor of Engineering from the University of New South Wales, and Master of Economics and Finance from Sydney University. He joined Southwestern University of Finance and Economics, China in 2023 as lecturer. His research interest includes Text Mining, Text Recommendation, NLP for Financial Technology, etc.



Dan Meng is a professor with the Southwestern University of Finance and Economics, Chengdu, China. Before that, she received Ph.D. degree from the Southwest Jiaotong University, Chengdu, China.Her research interests include Intelligent Finance, Intelligent Decision Making and Uncertainty Information Processing.



Jun Wang is a professor with the Southwestern University of Finance and Economics, China. Prior to taking that post, he was a researcher with the Memorial University of New-foundland at St. John's, Canada. He was awarded the National Scholarship in 2017. His research interests include NLP, social media, social network analysis, financial analysis, and business intelligence.



Jinghua Tan (Member, IEEE) received the B.S., M.S. and Ph.D. degrees from the Southwestern University of Finance and Economics, Chengdu, China. She is an associate professor with Sichuan Agricultural University. She was a visiting scholar with the Memorial University of Newfoundland at St. John's, Canada, in 2019. Her research interests include data mining and finance intelligence.