Discrete Mathematics, Algorithms and Applications

Surveying the Landscape of Text Summarization with Deep Learning: A Comprehensive Review

Guanghua Wang, Weili Wu

Computer Science Department, The University of Texas at Dallas, 800 W, Campbell Road Richardson, 75080-3021, United States
guanghua.wang@utdallas.edu
weiliwu@utdallas.edu

Received 7th August 2023
Revised 30th September 2023
Accepted 6th October 2023
Published Day Month Year

In recent years, deep learning has revolutionized natural language processing (NLP) by enabling the development of models that can learn complex representations of language data, leading to significant improvements in performance across a wide range of NLP tasks. Deep learning models for NLP typically use large amounts of data to train deep neural networks, allowing them to learn the patterns and relationships in language data. This is in contrast to traditional NLP approaches, which rely on hand-engineered features and rules to perform NLP tasks. The ability of deep neural networks to learn hierarchical representations of language data, handle variable-length input sequences, and perform well on large datasets makes them well-suited for NLP applications. Driven by the exponential growth of textual data and the increasing demand for condensed, coherent, and informative summaries, text summarization has been a critical research area in the field of NLP. Applying deep learning to text summarization refers to the use of deep neural networks to perform text summarization tasks. In this survey, we begin with a review of fashionable text summarization tasks in recent years, including extractive, abstractive, multi-document, and so on. Next, we discuss most deep learning-based models and their experimental results on these tasks. The paper also covers datasets and data representation for summarization tasks. Finally, we delve into the opportunities and challenges associated with summarization tasks and their corresponding methodologies, aiming to inspire future research efforts to advance the field further. A goal of our survey is to explain how these methods differ in their requirements as understanding them is essential for choosing a technique suited for a specific setting. This survey aims to provide a comprehensive review of existing techniques, evaluation methodologies, and practical applications of automatic text summarization.

Keywords: Natural language processing; Deep neural network; Text Summarization; Extractive Summarization; Abstractive Summarization; Multi-document Summarization

1. Introduction

1.1. *Overview of Deep Learning for Text Summarization*

Text summarization is the process of reducing a text or multiple texts to their essential meaning or main points while preserving its overall meaning and tone [106]. It has a wide range of applications, from creating news headlines and abstracts to summarizing legal documents and scientific papers. One common application of summarization is in news aggregation, where summaries of news articles are provided to users to consume news content quickly and efficiently [12,88]. Another important application of summarization is in the legal industry [55,77], where lawyers may need to quickly review large amounts of legal documents to identify relevant information. Summarization can also be used in healthcare to summarize medical records [3,6], which can help doctors and other healthcare professionals make more informed decisions. In the meanwhile, summarization is able to summarize social media posts for busy readers who want to stay informed but do not have time to read the entire document [33,122].

Traditional summarization models typically include rule-based ones or statistical techniques that focus on identifying key phrases and sentences from the source text without relying on deep learning or complex language models. Traditional methods have been widely used and provide a foundation for understanding the summarization task. One of them is the keyword-based method, which focuses on identifying keywords within the text and using them to select or rank sentences. Common techniques contain Term Frequency-Inverse Document Frequency (TF-IDF) weighting [32, 85], where sentences with a high concentration of important keywords are considered more relevant. Another approach is the Heuristic method, which relies on predefined rules or heuristics, such as considering sentence position, length, or similarity to the title, to determine the important sentences. For example, the Lead method [17,183] selects the first few sentences of a document, assuming that they contain the most critical information. Besides, graph-based systems represent the document as a graph, where nodes correspond to sentences and edges represent the relationships or similarities between them. Algorithms like PageRank [18,115] or LexRank [46] are used to identify the most important nodes (sentences) in the graph, which are then included in the summary. Latent Semantic Analysis (LSA) [139,164], on the other hand, is a statistical method that aims to capture the underlying semantic structure of a document by reducing its dimensionality. LSA is applied to a term-sentence matrix, and Singular Value Decomposition (SVD) is used to identify the most significant concepts or topics. Sentences that best represent these concepts are selected for the summary. The SumBasic algorithm [132] calculates the probability of a word appearing in a summary based on its frequency in the document. Sentences are scored by averaging the probabilities of their words, and the highest-scoring sentences are chosen for the summary. This method is simple but can yield reasonably good results.

However, the capacity of traditional methods to produce organized and smooth

summaries or adjust to diverse fields or dialects is frequently deficient. These methods are often simpler and faster than modern approaches, but they may not be as effective or accurate in capturing the nuances of the source text. Most of them are mainly focused on extractive summarization, which involves selecting the most important and relevant sentences or phrases from the original document to create a concise summary. Due to the complexity of generating new text, traditional methods are less common in abstractive summarization, which aims to generate a condensed version of the source text by rephrasing and reorganizing the original content instead of merely extracting existing sentences.

Nowadays, neural networks with multiple layers [31, 125, 161] enable the development of models that can understand, generate, and manipulate natural language from language data. Deep neural networks have demonstrated significant improvement in the performance of summarization tasks [39, 91], especially when compared to traditional statistical and traditional machine learning approaches. Deep learning models can learn from large amounts of data and generate more accurate predictions by capturing complex patterns and relationships in language data [151]. It can also handle the complexity and variability of natural language inputs, such as variable-length sequences of words and sentences. This allows the model to capture long-range dependencies and context, which are critical for understanding the meaning of a sentence or document. On the other hand, deep learning can also learn representations of language data end-to-end [167], without relying on hand-engineered features or rules. This approach enables the same model to be used for different tasks with minor modifications. With the large, general-purpose datasets [125] and high-performance computation ability [34] in recent years, deep learning can use pre-trained models as a starting point for a new summarization task with limited annotated data. The rapid advancements in deep learning with new architectures and techniques have led to a steady stream of innovations in summarization, which has pushed the state-of-the-art in language understanding and generation.

Numerous research papers have been published on the subject of text summarization in conjunction with deep learning. However, these papers vary in scope and focus: some primarily address prevalent models [5, 15, 64, 74, 103, 170], while others discuss the applications of summarization tasks [2, 58, 123, 190]. Some papers cover both aspects without delving into the datasets associated with text summarization [56, 128]. Additionally, certain papers only review a specific sub-field of summarization [4, 49, 66, 78, 97, 120, 130, 181]. This paper aims to provide a comprehensive overview of deep learning techniques for text summarization. This encompasses the key text summarization tasks and their historical context, widely adopted models, and beneficial techniques. Furthermore, a comparative analysis of performance across various models will be presented, followed by the prospects for their application. These aspects will be discussed in the subsequent sections.

1.2. *Paper Structure*

The rest of the paper is structured as follows:

- Section 2 presents a comprehensive review of different tasks and their brief histories in text summarization.
- Section 3 describes some of the most popular deep neural network models and related techniques.
- Section 4 reviews a recipe of datasets and shows how to quantify efficiency, and what factors to consider during the evaluation of each task.
- Section 5 discusses the main challenges and future directions for text summarization with deep learning techniques.

2. Tasks in Summarization

Summarization tasks can be classified into several categories based on different criteria, such as summarization method, source document quantity, source document length, summary length, and so on.

2.1. *Summarization Method: Extractive vs Abstractive*

Extractive and abstractive summarization are two primary approaches to text summarization. Extractive summarization aims to identify and select the most pertinent sentences or phrases from the original text [108], whereas abstractive summarization creates novel sentences that rephrase and consolidate the key concepts of the source document [42].

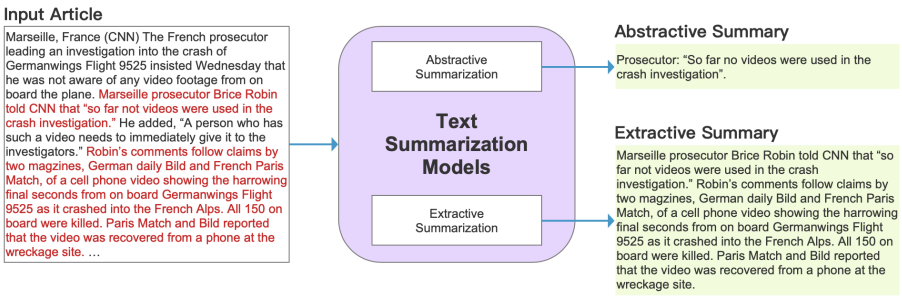

Fig. 1. The generated summary presented is the output of a "unilm-base-cased" model [11] that has been fine-tuned, whereas the extractive summary provided is the output of a "distilbert-base-uncased" model [157] that has also undergone fine-tuning. Both models were trained on the CNN/Daily Mail dataset. [38]

The underlying assumption of extractive summarization is that the original text contains sentences that are sufficiently informative and well-formed to be included in

the summary directly. Extractive summarization typically relies on techniques such as scoring sentences based on their relevance, importance, or position within the source text. A graph-based ranking model for text processing, called TextRank [115], is inspired by the PageRank algorithm used in web search. The model represents the input as a graph, where nodes are sentences or words, and edges represent the similarity between nodes. The algorithm iteratively scores nodes based on their connections, with higher-scoring nodes considered more important. Furthermore, LexRank [46], which is based on the concept of eigenvector centrality in a graph representation of the original text, is another unsupervised method for extractive summarization. On the other hand, Maximal Marginal Relevance (MMR) [23] addresses the problem of redundancy of extractive summaries by selecting sentences that are both relevant to the query and diverse from the already-selected sentences in the summary. The MMR algorithm iteratively selects sentences based on the combination of query relevance and novelty, considering the content of previously chosen sentences.

Instead of merely selecting existing sentences, abstractive summarization creates new sentences that convey the key ideas in a more natural and fluid manner. This approach requires a deeper understanding of the text and more advanced natural language generation capabilities. Goldstein et al. [59] propose a sentence extraction method based on a linear combination of several feature functions, followed by a sentence fusion step to generate abstractive summaries. Banko et al. [10] also use statistical models for content selection and surface realization to generate more succinct summaries. With probabilities calculated for candidate summary terms and their likely sequencing, a search algorithm is used to find a near-optimal summary in their system.

While extractive summarization can produce coherent and accurate summaries, it may be limited in terms of fluency and flexibility, as the selected sentences are directly taken from the source text and may not always fit together seamlessly. Abstractive summarization with traditional methods can produce more creative and tailored summaries but faces challenges of flexibility and expressiveness. With the advent of deep learning, neural network-based models like sequence-to-sequence (Seq2Seq) models [31], attention mechanisms [155], and transformers [91, 173] have significantly improved the performance of abstractive summarization by capturing complex patterns and semantic relationships within the original text, as well as extractive summarization.

2.2. *Source Document Quantity: Single-document vs Multi-document*

Single-document summarization and multi-document summarization are two distinct tasks within text summarization domain based on source document quantity. Single-document summarization focuses on generating a summary from a single input document, while multi-document summarization aims to create a summary by

aggregating information from multiple related documents [41, 111, 149]. Within the single-document system, the objective is to condense the main ideas and essential information contained in that specific document. On the other hand, multi-document summarization tasks require identifying and combining the most relevant and non-redundant information from a set of documents, often covering the same topic or event. This means multi-document summarization has additional challenges, such as maintaining cross-document coherence, efficiently handling larger volumes of information, and processing redundancy across documents. These complexities make multi-document summarization generally more difficult than single-document summarization.

Traditional approaches usually employ extractive techniques both in the context of single-document and multi-document summarization. Graph-based methods can be applied to both single-document and multi-document summarization by representing the relationships between sentences in one document or several documents as a graph, with sentences as nodes and the edges as the similarity between the sentences. The systems [46, 96, 141, 171] then use algorithms like PageRank, HITS, or LexRank to identify the most important sentences in the graph, which are then extracted and combined to form the summary.

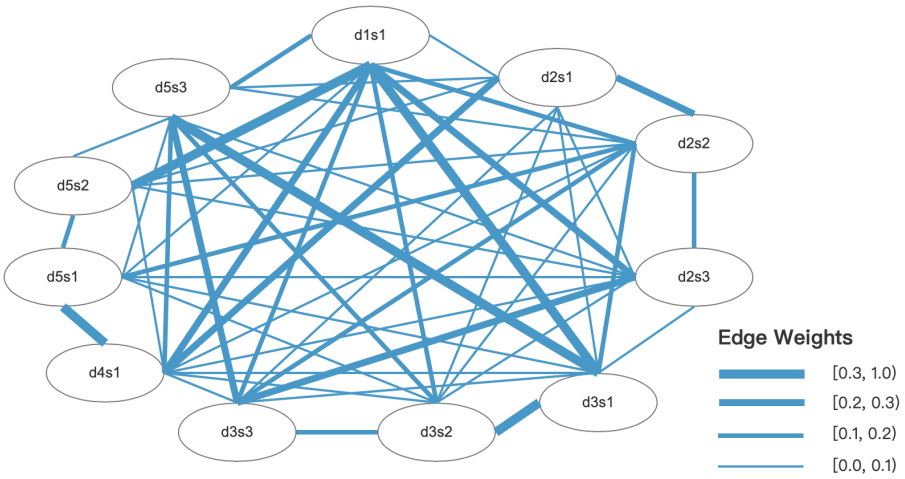

Fig. 2. The weighted cosine similarity graph [46] was generated for the cluster, based on the subset of d1003t from DUC 2004, which is a dataset for multi-document summarization tasks. The notation used in the figure is as follows: 'd' represents document and 's' represents sentence. For instance, d2s3 denotes the third sentence of document 2.

For single-document summarization, position-based methods [44, 82, 95] exploit the position of sentences within the document to identify important content. Additionally, the TF-IDF approach [60, 115, 156] weighs the importance of words in

the document based on their frequency and rarity. Sentences with high TF-IDF scores, which indicate a higher presence of significant words, are considered more important and are selected for the summary. Besides, Latent Semantic Analysis (LSA) [60,139,164] is a mathematical technique that reduces the dimensionality of the term-document matrix and uncovers the underlying semantic structure of the document. By identifying the principal components or latent topics, LSA can rank sentences according to their relevance to these topics, and the top-ranked sentences are extracted for the summary.

Regarding the task of multi-document summarization, Centroid-based methods [45, 149] calculate a central point for each document by considering the average term frequency of words in the document. Then centroids of all documents are used to compute an overall centroid, and sentences from different documents that are closest to this overall centroid are selected for the final summary. In the clustering [57, 112, 178] approach, the documents are grouped into clusters based on their similarity to represent a common theme or topic. Sentences from each cluster are selected as representatives, based on features in the document. The final summary is generated by concatenating these representative sentences from each cluster. MMR [23] is another technique that balances the relevance of the extracted information to the query and the diversity of the information to avoid redundancy in the summary of multi-document.

Recently, deep learning techniques have made significant advancements in both single-document and multi-document summarization. They employ various architectures, such as RNNs [22,124], Transformers [102,200], and pre-trained language models [99, 189], to generate coherent and informative summaries from single or multiple documents.

2.3. *Source Document Length: Short vs Long*

Short document summarization and long document summarization are two categories of text summarization, which differ based on the length and complexity of the input documents. Short document summarization focuses on generating summaries for relatively shorter documents, such as news articles [63,125], blog posts [70], or single research papers [20]. Due to the limited length of the input, short document summarization often requires less context and background knowledge to produce coherent summaries. The main challenge is to effectively identify the most important information and convey it in a concise manner while maintaining coherence and readability. Meanwhile, long document summarization deals with generating summaries for more extensive and complex documents, such as books [81, 199], lengthy reports [71], or collections of research papers [35]. The primary challenge is to capture the overall theme and essential details while managing a large volume of information. This often needs advanced techniques to handle and process lengthy texts, maintain coherence, and produce summaries that effectively convey the critical points. Presently, a benchmark dataset whose source document length averages

over 3,000 lexical tokens can be classified as long documents, given that most current state-of-the-art summarization systems (e.g., pre-trained models) are restricted to processing only 512 to 1,024 lexical tokens [13]. These constraints cannot be easily overcome without new techniques that aid in enabling current architectures to process extensive textual inputs [78].

Before the rise of deep learning, short document summarization primarily relied on algorithms like TF-IDF [115], centroid [45], LSA [164], and graph-based models [171], which are also used for the single-document summarization task discussed earlier. Due to limitations in system and algorithm capabilities, long document summarization was largely neglected until recent years. However, the emergence of deep neural networks has led to advancements in long document summarization task [13, 35, 61, 188], as well as improvements to short document summarization [39, 91, 157].

2.4. *Summary Length: Headline vs Short vs Long*

The output summary length of summarization tasks can vary, depending on the desired level of detail and the intended use case. Based on summary length, there are three distinct text summarization tasks, which include headline summarization, short summary summarization, and long summary summarization.

The aim of headline summarization is to generate a very brief and concise summary that captures the main theme or topic of the source text [92, 165, 206, 207]. The output is usually a single sentence or a short phrase. Headline summarization is often used for news articles, where the goal is to provide readers with an immediate understanding of the main topic or event without diving into the details. This type of summarization task requires the model to extract or generate the most crucial information and convey it in a limited number of words.

Short summary summarization [36, 125, 197] aims to produce a slightly longer summary that provides more context and details than a headline. Short summaries typically consist of a few sentences or a short paragraph. These summaries are useful for readers who want a quick overview of the source text without reading the entire document. Short summarization tasks require the model to identify and extract key points, main ideas, and essential information while maintaining the overall coherence and informativeness of the input text.

The target of long summary summarization task [107, 113, 197] is to generate more comprehensive summaries that cover a wider range of topics, subtopics, or details from the source text. These summaries can be several paragraphs or even longer, depending on the length and complexity of the original document. This sort of task is suitable for situations where readers want to gain a deeper understanding of the source material without reading it in its entirety. It needs the model to not only extract key information but also maintain the logical structure and relationships between different ideas, making it a more challenging task.

To produce concise summary headlines, Banko et al. [10] present a traditional

approach that can generate summaries shorter than a sentence by building statistical models for content selection and surface realization. The approach is similar to statistical machine translation and uses statistical models of term selection and term ordering. Content selection and summary structure generation can be combined to rank possible summary candidates against each other using an overall weighting scheme. The Viterbi beam search [72] can be used to find a near-optimal summary. For short summary summarization, a trainable summarization program based on a statistical framework [82] was developed, focusing on document extracts as a type of computed document summary. Features such as sentence length, fixed phrases, paragraph information, thematic words, and uppercase words are used to score each sentence. By employing a Bayesian classification function, the paper estimates the probability that a sentence will be included in a summary.

In recent times, deep learning methods for headline summarization and short summary summarization, include sequence-to-sequence models, attention mechanisms [169], and transformers like BERT [39], GPT [150], and T5 [152], have emerged. These models are trained on large corpora to generate concise and informative headlines by learning the most important and relevant information within the input documents. Long summary summarization, along with long document summarization tasks, was overlooked for a significant period of time due to limitations in hardware and algorithm capabilities. Nevertheless, the development of deep neural networks has brought about significant progress in the field of long summary summarization techniques [107, 113], like long document summarization tasks.

2.5. *Language: Single-Language vs Multi-Language vs Cross-Lingual*

Summarization tasks can be categorized based on the language involved, resulting in single-language, multi-language, and cross-lingual summarization. In single-language summarization, both the input documents and the generated summaries are in the same language. This is the most common type of summarization task, and the majority of research has focused on this area. Multi-language summarization involves generating summaries for documents in various languages, but the output summary is in the same language as the input document. For instance, if the input is in French, the summary will be in French, if the input is in Japanese in the same model, the summary will be in Japanese [142]. Cross-lingual summarization refers to the task of generating a summary in a target language for a source document written in a different language [90, 134]. This type of summarization task requires models to not only understand and extract the main ideas from the source document but also translate the extracted information into the target language.

Traditional methods such as keyword extraction [65, 147], Hidden Markov Model [54], and graph-based algorithms [7] were commonly used for multi-language sum-

Table 1. One example of cross-lingual summarization involves generating summaries of the same content in multiple languages [8].

Source text in English	Crude oil futures climbed 2% on Friday to a 28-month high, as the United States and Russia are in a deadlock over the Syrian issue, related concerns intensified. In October, the New York Mercantile Exchange's light sweet crude oil futures settlement price rose 2.16 US dollars to 110.53 US dollars per barrel.
Source text in Chinese	原油期货周五攀升*2%*，至*28*个月高点，因美俄两国在叙利亚问题上陷入僵局，相关担忧愈演愈烈。纽商所十月轻质低硫原油期货结算价涨*2.16*美元，至每桶*110.53*美元。
Summary in English	Oil prices hit a 28-month high as tensions in Syria escalated.
Summary in Chinese	油价创*28*个月新高因叙利亚紧张局势升级。

marization before the widespread adoption of deep learning. These methods were adapted to work across various languages and often employed language-specific resources such as stop-words and stemming tools [142] or language-agnostic features like TF-IDF [65] to improve generalization. Cross-lingual summarization, on the other hand, typically depends on machine translation or bilingual dictionaries to comprehend and extract content from the source language, followed by producing a summary in the target language. Two pipeline methods were commonly employed: one approach is first summarizing the source document and then translating the summary to the target language [16, 90, 176, 191, 195], while the other approach entails translating the source document to the target language and then generating a summary [134, 177].

Nowadays, both multi-language and cross-lingual summarization tasks have benefited from advanced deep-learning techniques. These deep neural models [29, 43, 135, 204] can learn representations for multiple languages in a shared embedding space, which makes it possible to perform multi-language and cross-lingual transfer learning. By fine-tuning these models, researchers have achieved impressive results in generating summaries across different languages without the need for extensive parallel data or explicit translation.

2.6. *Domain: General vs Specific domain*

General summarization pertains to the process of generating summaries for any type of text or domain, without focusing on any particular subject matter. Conversely, specific domain summarization is designed to produce summaries for texts within a specific domain or subject matter, such as legal documents, scientific articles, or news stories.

Specific domain summarization tasks typically demand domain-specific knowledge and may integrate specialized language models, ontologies, or rules to better capture the nuances and significant aspects of the target domain [26, 83, 89, 192]. These systems may also consider the distinctive structure or format of texts in the domain, leading to better summarization results. Prior to deep learning, tradi-

tional specific domain summarization methods often incorporated domain-specific knowledge in the form of rules [48, 172], or templates [138, 182, 208]. Additionally, feature-based methods were employed to identify important sentences or segments in the domain-specific summarization, such as the occurrence of certain keywords, phrases, or named entities relevant to the domain [27, 146, 148]. Graph-based algorithms, such as LexRank [46] or TextRank [115], were also commonly utilized for specific domain summarization. Domain-specific information could be incorporated into the graph representation or used to weigh the edges, making the algorithms more suited to the targeted domain.

With the advent of deep learning, specific domain summarization has significantly evolved and improved. Domain adaptation is a technique that leverages pretrained language models [83, 192, 201], which have been trained on vast amounts of text data and fine-tunes them for specific domain summarization tasks. To better capture the domain-specific knowledge, deep learning models also can be trained with domain-specific word embeddings or contextualized embeddings, such as BioBERT [89] for the biomedical domain or Legalbert [26] for the Legal documents. These techniques offer several advantages over traditional approaches, including better representation learning, more effective handling of domain-specific knowledge, and the ability to adapt to new domains more easily. They have shown promising results in various specific domains and continue to push the boundaries of what is possible in domain-specific text summarization.

2.7. *Level of Abstraction: Generic vs Query-focused*

Query-focused summarization aims to generate a summary that addresses a specific user query or topic. The summary should contain the most relevant information from the source text with respect to the given query [1, 25, 203]. This type of summarization is useful for readers who have a specific question or interest and want a summary tailored to their needs. In contrast, the goal of generic summarization is to create a concise and coherent summary that captures the most important information from the source text. This type of summarization is not focused on any specific query or topic. Instead, it aims to provide an overview of the entire document or set of documents, which can be helpful for readers who want to quickly grasp the key points without going through the entire text.

For query-focused, the relevance of the information is determined by its relation to the given query. Important information in a generic summary may be excluded if it is not directly relevant to the query, while less significant information related to the query may be included. Therefore, query-focused summarization typically requires additional processing to account for the given query. This may involve incorporating the query into the representation of the text, using query-specific features, or applying query-based attention mechanisms to guide the extraction or generation of the summary. These techniques are more specialized and require a query as input along with the source text.

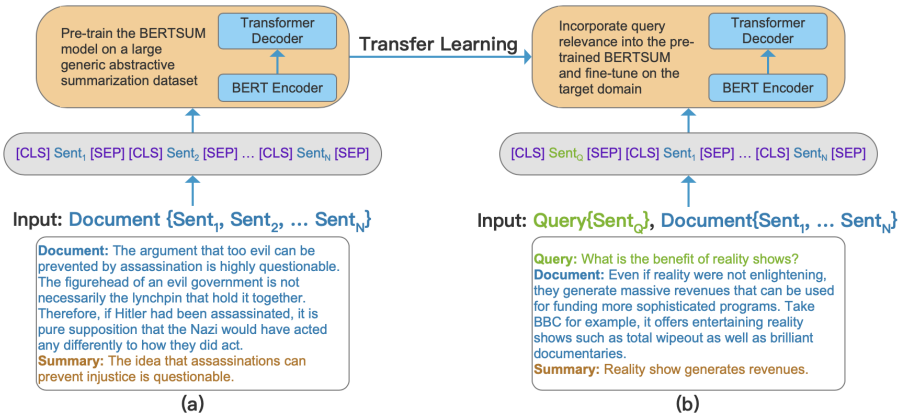

Fig. 3. (a) Pre-training the BERTSUM [101] model on a generic abstractive summarization corpus, such as XSUM [126]. (b) Fine-tuning the pre-trained BERTSUM model on the target domain, which in this case is Debatepedia [84, 129].

One of the simplest traditional approaches to query-focused summarization is to match the keywords in the query with those in the source text [37, 203]. The sentences containing a higher number of matched keywords are considered more relevant and are included in the summary. Besides, query expansion methods [24, 25] expand the initial query using various techniques, such as synonym extraction or related term discovery, to improve the recall of relevant information. The expanded query is then used to match and rank sentences in the source text. This method helps to identify relevant information that may not have been captured by the original query. Furthermore, query-based weighting techniques incorporate the query by utilizing term weighting schemes like TF-IDF [37,51]. In these methods, query terms are given higher weights, which enhances the relevance score of sentences containing those terms. The sentences are then ranked based on their scores, and the top-ranked sentences are selected for the summary. Graph-based algorithms, like LexRank [46] or TextRank [116], can also be adapted for query-focused summarization by incorporating the query into the graph representation of the text. On the other hand, supervised machine learning like Support Vector Machines (SVM) [52, 163] or logistic regression [137], can be trained to rank the sentences in new, unseen documents based on their relevance to the query for query-focused summarization.

With the appearance of deep learning techniques, such as Seq2Seq models [175], pointer generator networks [67], pre-trained language models [1], and reinforcement learning [21], query-focused summarization could better understand the input and query context to generate more accurate and relevant summaries that address the specific user query or topic.

3. Deep learning techniques

This section will review the most popular deep learning models and techniques utilized with deep neural networks, such as attention mechanisms, copy mechanisms, dictionary probabilities, etc. Each type of model has the potential to significantly improve various summarization tasks, which may be a blend of different categories, such as long abstractive legal document summarization, cross-lingual headline summarization, and extractive query-focused summarization.

3.1. *Plain neural network*

Plain neural network, also known as feed-forward neural network [14, 86], is a type of artificial neural network that consists of several layers of interconnected nodes or neurons. The input is passed through one or more hidden layers, where the weights of each neuron are adjusted based on the error generated by the network's output. This process, called back-propagation, allows the network to learn how to classify or predict outputs based on the input data.

Plain neural network is usually used in learning vector representation of words or sentences in summarization tasks. Word2vec models [117, 118], which learns continuous vector representations of words from large amounts of text data, was developed with plain neural network. After these models, Glove [144] was proposed to combine the advantages of matrix factorization and local context window methods to create a more efficient and accurate model. The learned word vectors can capture various semantic and syntactic regularities, and can be used as features for different summarization tasks. On the other hand, Paragraph Vector [68] learns fixed-length distributed representations of variable-length pieces of text by jointly predicting the words in the text and a separate paragraph-specific vector.

3.2. *Recurrent neural network*

Recurrent neural network (RNN) [154,159] is specifically designed to handle sequential data, making them highly suitable for NLP tasks. RNN can process variable-length sequences of inputs and maintain an internal state, which allows the model to remember information from previous time steps. However, RNN is prone to vanishing and exploding gradient problems, which makes it challenging to learn long-range dependencies.

To address these issues, Long Short-Term Memory (LSTM) [69] and Gated Recurrent Unit (GRU) [30] networks were developed to selectively remember or forget information over time. Each memory cell in an LSTM network has three main components: an input gate, a forget gate, and an output gate. The input gate determines how much of the new input information should be stored in the memory cell, while the forget gate determines how much of the previous memory state should be forgotten. The output gate controls how much of the current memory state should be used to generate the output. As opposed to LSTM, GRU uses a

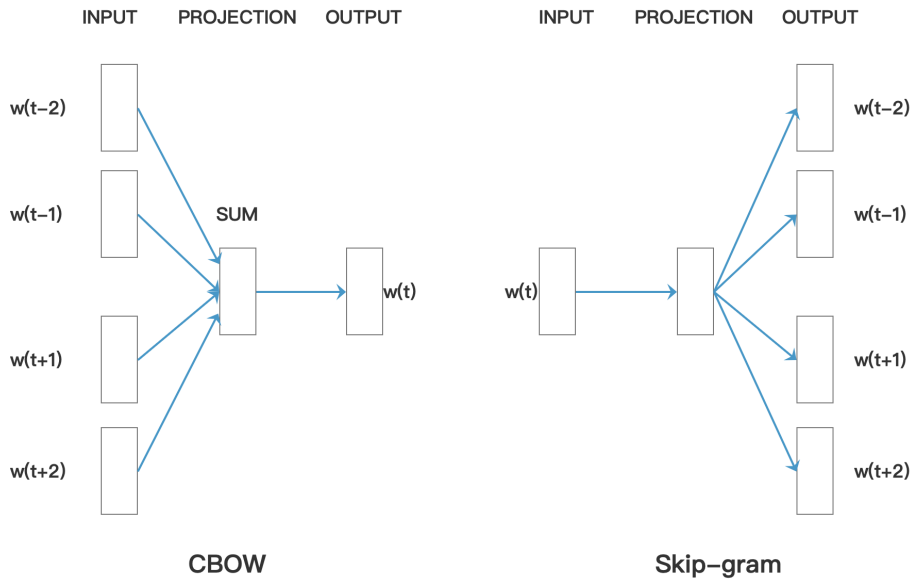

Fig. 4. The current word was predicted based on the context by CBOW, and the surrounding words were predicted based on the current word by the Skip-gram. [117]

simpler gating mechanism that has only two gates: an update gate controls how much of the previous memory state should be retained and how much of the new input should be added to the memory cell, a reset gate determines how much of previous memory state should be ignored in favor of the new input.

For extractive multi-document summarization, SummaRuNNer [124] is a GRU-based RNN model, which allows visualization of its predictions based on abstract features such as information content, salience, and novelty. It is an extractive model using abstractive training, which can train on human-generated reference summaries alone, removing the need for sentence-level extractive labels.

The attention mechanism is a technique that allows models to focus on different parts of the input when producing an output. Both Chopra et al. [31] and Nallapati et al. [125] have investigated the utilization of Attentional RNN Decoder for enhancing abstractive summarization performance. In the former study, a Convolutional attention-based network is employed as the encoder, furnishing a conditioning input to guide the decoder in focusing on relevant portions of the input sentence during word generation. On the other hand, the latter study employs an RNN-based encoder and incorporates keyword modeling to capture the hierarchical structure between sentences and words, effectively addressing the challenge of rare or unseen words. To tackle this issue, they propose a novel switching decoder/pointer architecture that enables the model to make decisions between generating a word and indicating its location within the source document.

Furthermore, See et al. [161] presents a hybrid pointer network that copies words from the source text to reproduce accurate information and handle out-of-vocabulary words. They also developed a coverage architecture to avoid repetition in the summary. This aids the attention mechanism to avoid repeatedly attending to the same locations, reducing the generation of repetitive text. Additionally, they utilize beam search, which is a heuristic search algorithm that explores the search space in a breadth-first manner to find the most likely output sequence. It extends the search to the top 'B' candidates at each step, where 'B' is a predefined beam width. The beam width determines the number of alternatives (branches) to explore at each step. At each time step, it keeps track of the top 'B' sequences based on the probabilities of the sequences. The final output sequence is the one that has the highest overall score. Beam search offers a good trade-off between the quality of output and computational efficiency.

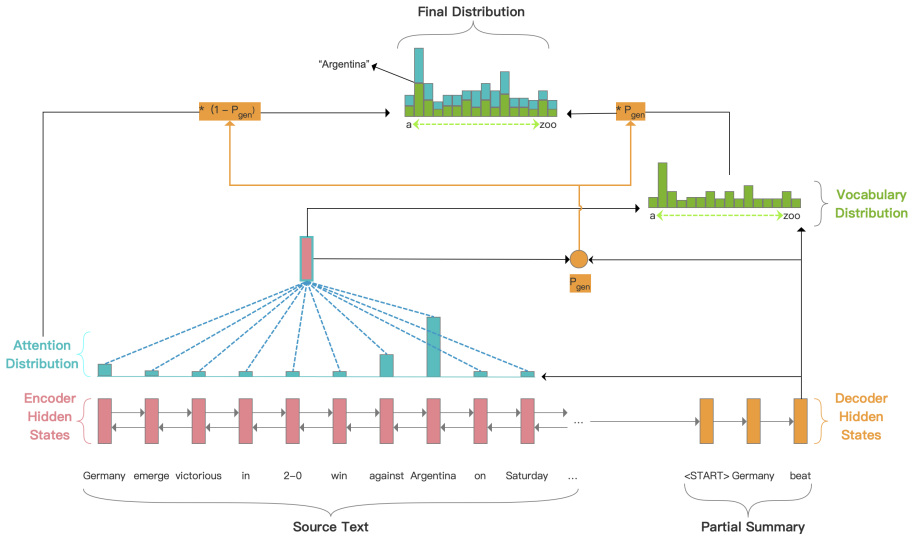

Fig. 5. Pointer-generator model combines an RNN with attention and copy network. [161]

Zheng et al. [205] propose a new method for multi-document summarization called Subtopic-Driven Summarization. The authors argue that each underlying topic of the documents can be seen as a collection of different subtopics of varying importance. The proposed model uses these subtopics and generates the underlying topic representation from a document and subtopic view. The goal is to minimize the difference between the representations learned from the two views. The model uses a hierarchical RNN to encode contextual information and estimates sentence importance hierarchically considering subtopic importance and relative sentence importance.

3.3. *Convolutional neural network*

In NLP, Convolutional neural networks (CNN) [53, 80, 87], which originally developed for image processing and computer vision tasks, are used to capture local patterns and structures within a text, making them particularly effective for tasks that involve extracting meaningful features from a sequence of words or characters. In a typical CNN architecture for NLP, the input text is first represented as a sequence of word or character embeddings, forming a matrix where each row corresponds to a word or character embedding. The embeddings are learned during the training process, allowing the model to capture meaningful representations of words or characters. The main building block of CNN is the convolutional layer, which consists of multiple filters. Each filter is applied to sliding windows of fixed size across the input text, capturing local patterns within the text. The filter's output, or feature map, is then passed through a non-linear activation function, such as the Rectified Linear Unit, to introduce non-linearity into the model. After the convolutional layers, the feature maps are typically passed through a pooling layer, such as max-pooling or average pooling, which reduces the spatial dimensions and extracts the most salient features from the feature maps. This process helps to reduce the computational complexity of the model and makes it more robust to variations in the input. The final layers of a CNN for NLP usually consist of one or more fully connected layers, which combine the extracted features and perform the specific NLP task, such as text classification or sequence tagging. These layers are often followed by a softmax layer for multi-class tasks to produce probability distributions over the possible output classes.

Narayan et al. [126] introduce a new concept called "extreme summarization", which aims to generate a single sentence summary that can answer the question "What is the article about?". The researchers proposed a novel abstractive model that is conditioned on the article's topics and based entirely on CNN. This model was found to effectively capture long-range dependencies and identify pertinent content in a document. They also incorporate topic-sensitive embeddings to enhance word context with their topical relevance to the documents.

Liu et al. [105] discuss a new approach for generating summaries of user-defined lengths using CNN. The proposed approach modifies a convolutional sequence-to-sequence model to include a length constraint in each convolutional block of the initial layer of the model. This is achieved by feeding the desired length as a parameter into the decoder during the training phase. At test time, any desired length can be provided to generate a summary of approximately that length. This research contributes a potentially effective method for producing summaries of arbitrary lengths, which holds promise for diverse applications across various tasks requiring summaries of different lengths.

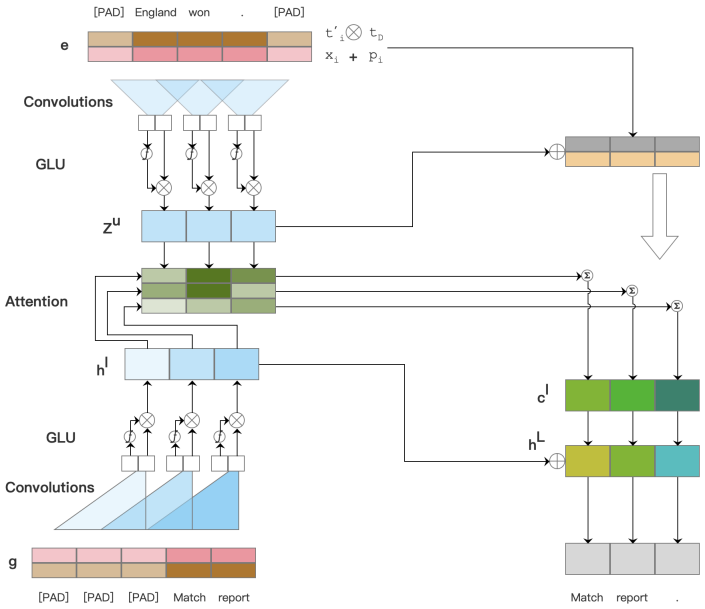

Fig. 6. A convolutional model conditioned on the topic for extreme summarization. [126]

3.4. *Graph neural networks*

Although Graph Neural Networks (GNN) [62, 114, 145, 158] are primarily used in domains where data naturally exhibits graph structures, such as social networks, molecular structures, and knowledge graphs, they can also be adapted for NLP tasks. In NLP, GNNs are typically used to model relationships between words, sentences, or documents by representing them as nodes in a graph, with edges representing the relationships between these nodes. A GNN model learns to propagate information through the graph by iteratively updating the node representations based on the information from their neighbors. The core building blocks of GNNs are graph convolutional layers, which are designed to aggregate information from neighboring nodes and update the node features.

Jing et al. [75] present a novel Multiplex Graph Convolutional Network (Multi-GCN) approach for extractive summarization. Multi-GCN learns node embedding of different relations among sentences and words separately and combines them to produce a final embedding. This approach helps to mitigate the over-smoothing and vanishing gradient problems of the original GCN.

A heterogeneous GNN, HETERSUMGRAPH [179] is introduced for extractive document summarization. This network includes nodes of different granularity levels apart from sentences, which act as intermediaries and enrich cross-sentence relations. This approach allows different sentences to interact considering overlapping word information. Moreover, the graph network can accommodate additional node

types, such as document nodes for multi-document summarization.

Doan et al. [40] propose a method for long document summarization by applying Heterogeneous Graph Neural Networks (HeterGNN) and introducing a homogeneous graph structure (HomoGNN). The HomoGNN focuses on sentence-level nodes to create a graph structure, enriching inter-sentence connections. Simultaneously, the HeterGNN explores the complex relationships between words and sentences, tackling intra-sentence connections. Both networks are constructed and updated using a Graph Attention Network model. In the HomoGNN, a BERT model is used for the initial encoding of sentences, while the HeterGNN uses a combination of CNN and BiLSTM for node feature extraction. After processing, the outputs of both networks are concatenated to form the final representation of sentences.

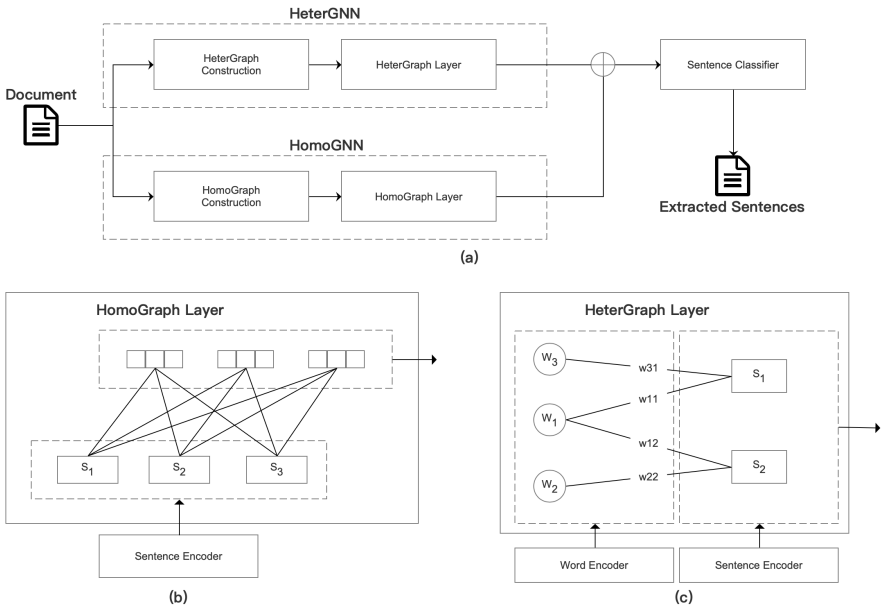

Fig. 7. A two-phase pipeline model for extreme summarization [40]. In the first phase, sentences are encoded using pre-trained BERT, and the [CLS] token information is passed through a graph attention layer. In the second phase, both word and sentence nodes are encoded and fed into a heterogeneous graph layer. The outputs from the two phases are concatenated and inputted into a multi-layer perceptron (MLP) layer for sentence label classification.

3.5. *Transformer*

Transformers models [39, 150, 173], which are the most popular deep learning architecture, have been a revolutionary force in the field of NLP, especially in text summarization. Unlike RNN or LSTM, Transformers use the self-attention mechanism, allowing them to capture dependencies regardless of their distance in the input

text. This is particularly useful in text summarization tasks, where understanding the full context of a document is crucial.

Transformer [173] follows a Seq2Seq architecture and consists of an encoder to map the input sequence into a latent space, and a decoder to map the latent representation back into an output sequence. At the heart of the Transformer model is the self-attention mechanism, which allows the model to weigh the significance of words in the input sequence when generating each word in the output sequence.

BERT [39], which is Bidirectional Encoder Representations Transformers, pre-trains deep bidirectional representations from the unlabeled text by conditioning on both left and right context in all layers. After pre-training, the BERT model can be fine-tuned with an additional output layer to create state-of-the-art models for a wide range of tasks, including summarization, without task-specific architecture modifications. During the fine-tuned phase, the model is initialized with the pre-trained parameters, and all parameters are fine-tuned using labeled data from the downstream tasks. To use BERT for text summarization, a common method is to fine-tune it on a summarization task. BERTSUM [101] is an approach to utilize BERT for extractive summarization by adding an interval segment embedding and a positional embedding to the pre-trained BERT model, allowing the model to recognize sentences and their orders. These embeddings are learned during the fine-tuning process. During inference, the most important sentences are selected based on their scores to form the final summary.

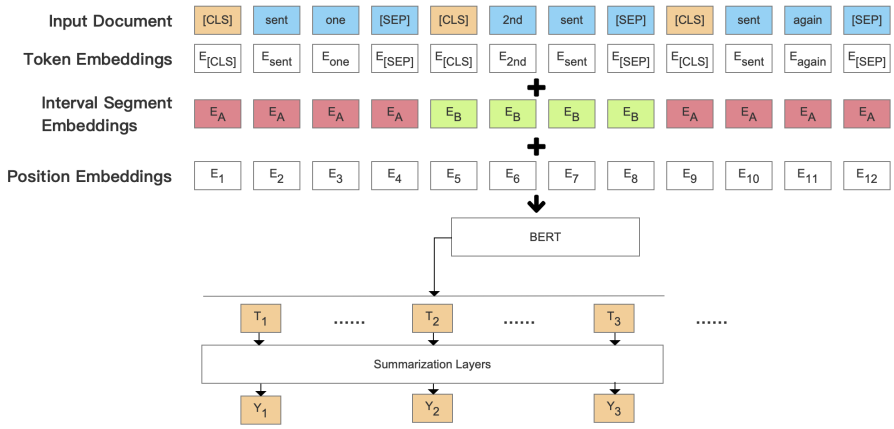

Fig. 8. The overview architecture of the BERTSUM model [101].

T5 [153], short for "Text-to-Text Transfer Transformer", is a unified model that treats every NLP problem as a text generation task, enabling the model to learn multiple tasks simultaneously and to learn shared representations across these tasks. In the case of text summarization, the model is trained to predict the summarized

text given the original text prefixed with a task-specific instruction, like "summarize:", so the model learns to generate the summary based on the context and the given task. T5 is trained using a denoising auto-encoding objective, which is essentially a causal language modeling task with some noise in the input data. The model has to learn to predict the original clean text from the noisy version. This method forces the model to learn to understand and generate grammatically correct and contextually relevant text, a skill that's very useful in generating coherent and relevant summaries.

BART (Bidirectional and Auto-Regressive Transformers) [91], is a denoising auto-encoder used for pertaining Seq2Seq models. The system works by corrupting text with an arbitrary noising function and then training the model to reconstruct the original text, using a standard Transformer-based neural machine translation architecture. This architecture generalizes the approach used with a bidirectional encoder and a left-to-right decoder and is particularly effective when fine-tuned for text generation tasks like summarization. Several different noising strategies were tested in BART, with the best performance achieved by randomly shuffling sentence order and using an innovative in-filling scheme, where segments of text are replaced with a single mask token. This forces the model to consider overall sentence length and make longer-range transformations to the input.

Pegasus [196], which stands for Pre-training with Extracted gap sentences for Abstractive Summarization, is a model that specifically focuses on abstractive text summarization. Pegasus's main novelty lies in its pre-training strategy, which simulates summarization by masking certain sentences in the document. Instead of masking individual words or phrases, Pegasus masks entire sentences, treating the task as a sentence-level extraction problem. During this phase, the model learns to predict the 'masked' sentences based on the rest of the text, developing a strong sense of sentence-level importance and relevance skills that are vital for text summarization. The model was tested on diverse domains including news, science, stories, and legislative bills, and showed strong performance on all tested datasets, as well as low-resource summarization.

BIGBIRD [194], a sparse attention mechanism designed to tackle the quadratic dependency of the sequence length, which is a limitation of Transformer-based models like BART. The BIGBIRD mechanism reduces this quadratic dependency to linear, meaning it can handle sequences up to 8 times longer than previously possible on the same hardware. This means that BIGBIRD can understand and generate much longer pieces of text, making it potentially useful for long document summarization. The model consists of three parts: a set of global tokens attending to all parts of the sequence, all tokens attending to a set of local neighboring tokens, and all tokens attending to a set of random tokens. BIGBIRD retains all the known theoretical properties of full transformers and extends the application of the attention-based model to tasks where long contexts are beneficial.

Longformer [13] is another modification of the Transformer model designed to handle long sequences. The Longformer offers a linearly scaling attention mecha-

nism to make it possible to process documents with thousands of tokens or more. Longformer's attention mechanism is a combination of local windowed self-attention and task-motivated global attention. This drop-in replacement for the standard self-attention could achieve state-of-the-art results on character-level language modeling tasks. Additionally, Longformer-Encoder-Decoder (LED) is introduced as a variant of Longformer designed for long document generative Seq2Seq tasks. This model is also tested and proven effective on the long document summarization dataset.

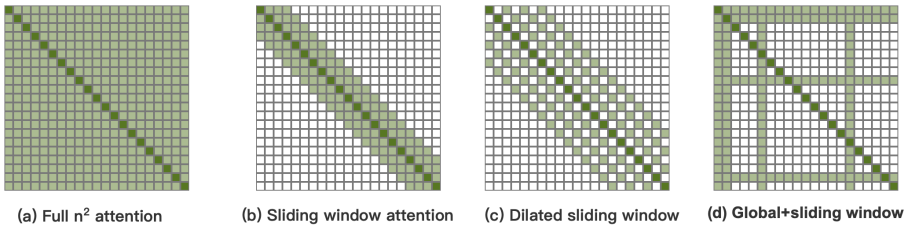

Fig. 9. By comparing the complete self-attention pattern and the attention pattern configuration in the Longformer model, we can observe the differences. [13].

GPT-1 [150] (Generative Pre-trained Transformer) is an auto-regressive language model, which can generate human-like by predicting the next word in a sequence. In the context of text summarization, GPT-1 is able to create summaries that are not just extracts of the original text but can rephrase or reframe the content in novel ways, capturing the essence of the document while potentially reducing its length significantly. However, since GPT-1 does not explicitly model the structure of the document beyond the sequence of words, it might not always maintain the coherence and relevance of the summary, especially for long or complex documents. GPT-2 [151] is an improvement over GPT-1, having more parameters and trained on a larger corpus of data. One important advantage of GPT-2 for summarization is its ability to generate fluent and coherent text due to its training objective and architecture. It can capture long-range dependencies in the text, rephrase the original text, and even generate novel sentences that were not in the original document but are consistent with its content. Furthermore, GPT-2 can be employed for summarization in a few-shot/zero-shot manner where the model is designed to make accurate predictions given only a few or no examples. GPT-3 [19] follows the design of its predecessor and has been found to generate exceptionally fluent and coherent text based on a larger model size- 175 billion parameters. GPT-3 has a larger context window, meaning it can consider a significant portion of a document when generating a summary. This allows for more holistic and comprehensive summaries, especially when compared to models with smaller context windows that might not capture all necessary information. Apart from the typical summarization tasks, GPT-3 can be leveraged for a range of different summary

types. Whether you're interested in producing extractive or abstractive summaries, single-document or multi-document summaries, GPT-3 can be utilized to generate them. InstructGPT [136], a recent development, centers around training large language models to comprehend and follow instructions with the aid of human feedback. The authors employ a fine-tuning approach on GPT-3, utilizing a dataset of labeler demonstrations that outline the desired model behavior, starting from labeler-written prompts and responses. They initially fine-tuned the model using supervised learning and subsequently employed reinforcement learning techniques with human feedback for further fine-tuning. The authors' findings indicate that fine-tuning models with human feedback hold promise in aligning language models with human intent, highlighting a fruitful direction for future research. Recently, a substantial multi-modal model known as GPT-4 [133] has emerged. It has the ability to process both image and text inputs and generate text outputs. In human exam evaluations, this model demonstrates exceptional performance, consistently outscoring the majority of human test takers. Despite the lack of specific information regarding the model's structure, hyper-parameters, and training methodology in the paper, the results of GPT-4 exhibit remarkable advancements across diverse tasks, including summarization. Even though GPT-4 might introduce details or points that were not part of the original document, leading to "hallucinations", the development of this model marks a significant milestone in the advancement of AI systems that are both widely applicable and safely deployable.

3.6. *Reinforcement learning*

Reinforcement Learning (RL) [76,119] is a type of Machine Learning where an agent learns to behave in an environment, by performing certain actions and receiving rewards (or punishments) in return.

In the context of text summarization, the environment consists of the input document that needs to be summarized. The state could be the current part of the document being considered for summarization, along with the portion of the summary that has already been generated. The action might involve selecting a sentence from the document to include in the summary (for extractive summarization) or generating a sentence or phrase to add to the summary (for abstractive summarization). The reward is a measure of the quality of the generated summary. This could be based on a variety of factors, such as how well the summary represents the main points of the document, how grammatically correct and fluent it is, and so on.

One of the key benefits of using RL for text summarization is that it allows for a more flexible and adaptive approach to summarization, compared to traditional supervised learning methods. RL can learn to adapt its summarization strategy based on the specific characteristics of each document and can optimize for long-term rewards (like the overall coherence and quality of the summary), rather than just short-term gains (like the accuracy of the next sentence).

Narayan et al. [127] proposes a novel method for single document summarization

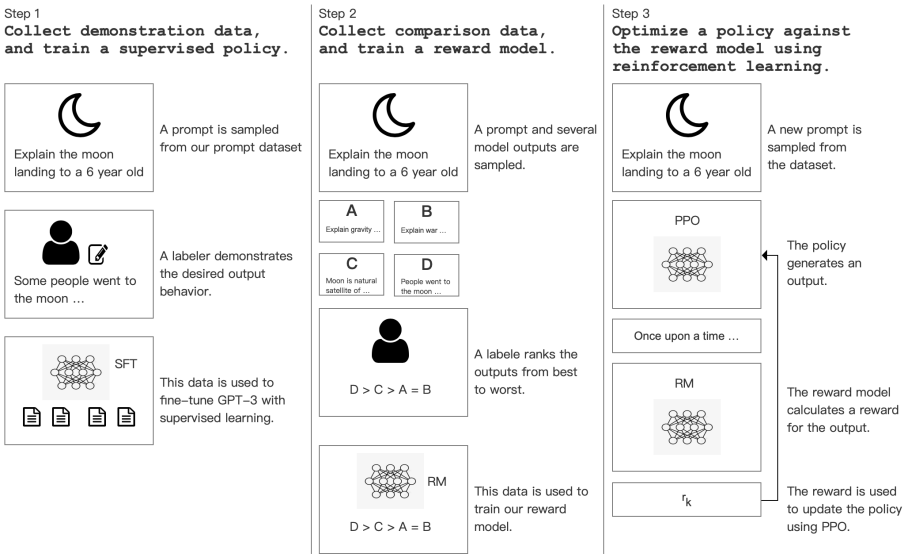

Fig. 10. The diagram showcases the three sequential steps of InstructGPT: (1) supervised fine-tuning (SFT), (2) reward model (RM) training, and (3) reinforcement learning through proximal policy optimization (PPO) using this reward model. The blue arrows indicate the utilization of this data to train InstructGPT. [136].

using extractive summarization as a sentence ranking task, globally optimizing the ROUGE evaluation metric through a reinforcement learning objective. The authors argue that current cross-entropy training is sub-optimal for extractive summarization, tending to generate verbose summaries with unnecessary information. Their method improves this by learning to rank sentences for summary generation. The approach involves viewing the neural summarization model as an "agent" in a reinforcement learning paradigm, which reads a document and predicts a relevance score for each sentence. The agent is then rewarded based on how well the extract resembles the gold-standard summary. The REINFORCE algorithm is used to update the agent, optimizing the final evaluation metric directly instead of maximizing the likelihood of the ground-truth labels, making the model more capable of discriminating among sentences.

Hyun et al. [73] introduce another model for unsupervised abstractive sentence summarization using reinforcement learning (RL). Unlike previous methods, which mainly utilize extractive summarization (removing words from texts), this method is abstractive, allowing for the generation of new words not found in the original text, thereby increasing flexibility and versatility. The approach involves developing a multi-summary learning mechanism that creates multiple summaries of varying lengths from a given text, with these summaries enhancing each other. The RL-based model used assesses the quality of summaries using rewards, considering fac-

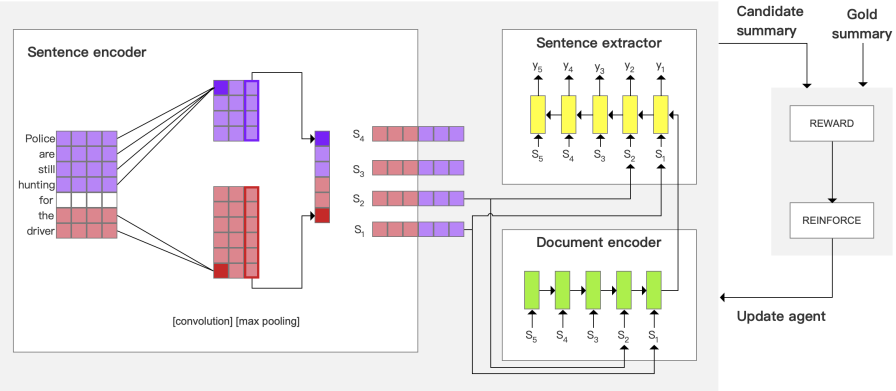

Fig. 11. The extractive summarization model with reinforcement learning [127] employs a hierarchical encoder-decoder architecture to rank sentences based on their extract worthiness. A candidate summary is then formed by assembling the top-ranked sentences. The REWARD generator assesses the candidate summary against the gold summary, providing a reward signal that is utilized in the REINFORCE algorithm [185] to update the model.

tors such as the semantic similarity between the summary and the original text and the fluency of the generated summaries. The model also involves a pre-training task to achieve well-initialized model parameters for RL training.

4. Data and Experimental Performance Analysis

4.1. *Techniques for data processing*

This sub-section will delve into the key techniques utilized for data processing.

4.1.1. *Pre-training*

In the domain of text summarization, pre-training refers to the initial training phase of a model on a large, diverse corpus of text data, prior to fine-tuning it on a more specific summarization task. This strategy capitalized on the capabilities of extensive language models such as GPT [150], BART [91], and T5 [153], which have been pre-trained on vast quantities of text data to comprehend syntactic and semantic patterns within a language.

4.1.2. *Few-shot, zero-shot learning*

Few-shot and zero-shot learning are terms used to describe scenarios where a model is required to make accurate predictions for new classes that were either minimally represented (few-shot) or completely absent (zero-shot) during the training phase [166,187]. Few-shot learning in summarization usually implies a scenario where the model is trained on many examples from a few categories and is then expected to

generalize to summarizing examples from new categories after seeing only a few examples from these new categories. Zero-shot learning in summarization, on the other hand, refers to a scenario where the model is expected to generalize to entirely new categories without seeing any examples from these categories during training. The idea behind these methods is to provide the model with a few examples or a description of the task at inference time, allowing it to adjust its predictions based on this new information. This often involves formulating the summarization task as a type of prompt that the model is designed to complete.

4.1.3. *Prompting*

Prompts play a crucial role in the current generation of language models, particularly those that are trained in a transformer-based architecture like GPT-4 [133] or T5 [153]. The term "prompt" in the context of these models refers to the input given to the model to indicate the task it should perform. For text summarization, the prompt is typically the text that needs to be summarized. However, in the case of GPT-4 [133], T5 [153], and similar models, the prompt can also include a task description or examples to guide the model's generation. This is especially important in few-shot and zero-shot learning scenarios. Choosing effective prompts is a bit of an art and can significantly impact the performance of the model. The best practices for creating prompts are still an active area of research, but a well-designed prompt often includes clear instructions and, when possible, an example of the desired output.

4.1.4. *Domain adaptation*

Domain adaptation in the context of text summarization pertains to the process of adapting a summarization model, initially trained on a specific domain (e.g., news articles), to perform effectively on a different yet related domain (e.g., scientific papers or legal documents) [110, 121]. Fine-tuning the model on a smaller dataset from the target domain is a commonly employed approach. Additionally, transfer learning is utilized to capitalize on the knowledge acquired from one domain and apply it to another.

4.1.5. *Tokenization, embedding and Decoding strategies*

Tokenization is the initial step of dividing the text into individual units known as tokens, which serve as fundamental input elements for most natural language processing (NLP) models [109,184]. Following tokenization, the tokens are converted into continuous vector representations through an embedding layer to create the input embeddings [144]. These embeddings are then fed into the model for further processing. Once the model generates an output sequence, such as a summary, the reverse process takes place, where each token is mapped back to its corresponding

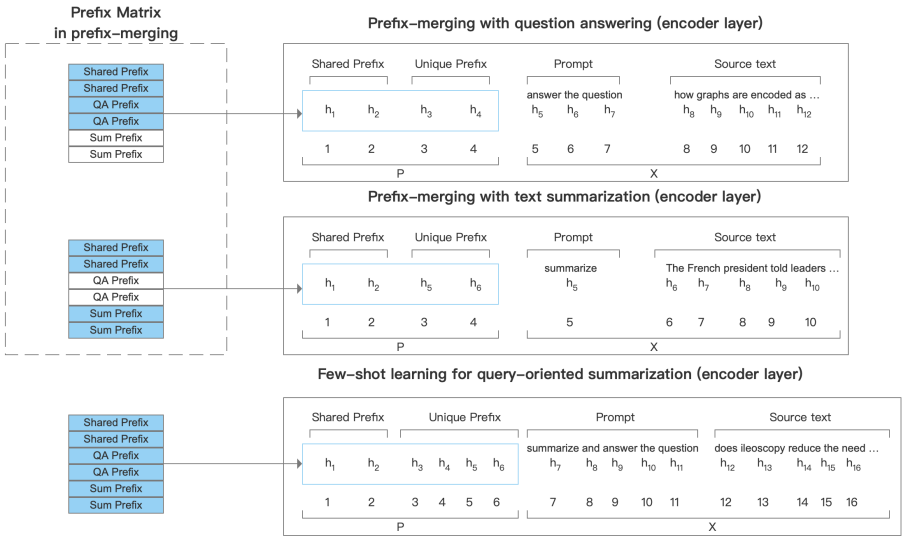

Fig. 12. The figure highlights the encoder layer of BART and provides annotated examples and a comparison between prefix-merging on the two auxiliary tasks (top, mid) and the application of the merged prefix on the Few-shot Query-Focused Summarization task using prefix-tuning (bottom). [193]

word in a vocabulary, and the words are subsequently joined together using spaces to produce human-readable text.

4.2. *Dataset and Evaluation Metrics*

In the field of text summarization, various datasets and evaluation metrics are used to train models and assess their performance. Each evaluation metric has its strengths and weaknesses. While some metrics are easy to compute, they might not accurately reflect the human judgment of quality as they are based solely on n-gram overlap. Some other metrics attempt to address this issue by taking semantic similarity into account. However, manual evaluation by human judges is still considered the gold standard, despite its scalability challenges. We will first discuss some commonly used datasets, and then we'll talk about evaluation metrics.

4.2.1. *Dataset*

Table 2 presents a compilation of notable datasets in the field of text summarization. The "Size" column indicates the respective counts of training, validation, and test documents available in each dataset.

Table 2. A comparison of different Datasets

Dataset	Domain	Tasks	Size	Document Length	Summary Length
DUC2004 [36]	News	single-document multi-document cross-lingual query-focused Abstractive	500	-	<= 75 bytes (single) <= 665 bytes (multi)
CNN/Daily Mail [125]	News	single-document Headline query-focused Abstractive Extractive	286,817 13,368 11,487	766 words average	53 words average
XSum [126]	News	single-document cross-lingual Abstractive	204,045 11,332 11,334	431 words average	23 words average
WikiSum [100]	Wiki	multi-document Abstractive	1,865,750 233,252 232,998	-	-
Multi-News [47]	News	multi-document Abstractive Extractive	44,972 5,622 5,622	2,103 words average	263 words average
BillSum [79]	Legal	single-document Long-document Abstractive	18,949 1,237 3,269	1,592 words average	197 words average
PubMed [35]	Medical	single-document Long-document Abstractive	119,924 6,633 6,658	3,016 words average	203 words average
arXiv [35]	Scientific	single-document Long-document Abstractive	203,037 6,436 6,440	4,938 words average	220 words average
XGLUE [93]	News	cross-lingual Headline Abstractive	300,000 50,000 50,000	-	-
BIGPATENT [162]	Patent	single-document Long-document Extractive Abstractive	1,207,222 67,068 67,072	3,572 words average	116 words average
Newsroom [63]	News	single-document Headline Extractive Abstractive	995,041 108,837 108,862	658 words average	26 words average
MLSUM [160]	News	single-document cross-lingual Multi-lingual Abstractive	287,096 11,400 10,700	790 words average	55 words average

4.2.2. *Evaluation Metrics*

Evaluating the quality of generated summaries is crucial in the text summarization task. Here is an assortment of evaluation metrics commonly employed in summarization:

- Rouge (Recall-Oriented Understudy for Gisting Evaluation) [94]: is a set of evaluation metrics used for evaluating automatic summarization. It compares the system-generated output with a set of reference summaries. ROUGE-N measures the overlap of N-grams (a contiguous sequence of N items from a given sample of text or speech) between the system and reference summaries. It includes metrics like ROUGE-1 (for unigrams), ROUGE-2 (for bigrams), and so on. ROUGE-L metric measures the Longest Common Subsequence (LCS) between the system and reference summaries. LCS takes into account sentence-level structure similarity naturally and identifies the longest co-occurring in sequence n-grams automatically.
- BLEU (Bilingual Evaluation Understudy) [140]: is an evaluation metric initially developed for assessing the quality of machine-translated text, but it has also been used in text summarization tasks. It is a precision-based metric that compares the system-generated summary with one or more reference summaries. BLEU operates at the n-gram level to measure the overlap of n-grams between the generated output and the reference texts. It calculates the precision for each n-gram size (usually from 1-gram to 4-gram) and takes a weighted geometric mean to compute the final score.
- METEOR (Metric for Evaluation of Translation with Explicit Ordering) [9]: is an evaluation metric initially designed for machine translation tasks but also used in text summarization evaluations. Unlike the previously mentioned metrics like ROUGE and BLEU that mainly focus on recall and precision at the n-gram level, METEOR incorporates more linguistic features and tries to align the generated text and the reference at the semantic level, thus potentially capturing the quality of the output more accurately.
- Pyramid Score [131]: is based on the principle that a perfect summary could include any of several valid points from the source text, and as such, it would not be fair to penalize a summary for not including specific points. In the Pyramid Score method, human assessors identify Summary Content Units (SCUs) in a set of model summaries, which are essentially nuggets of information. Each SCU is assigned a weight based on how many model summaries it appears in. The Pyramid Score is then computed for a system-generated summary by adding up the weights of the SCUs it contains and normalizing this sum by the maximum possible score achievable by any summary of the same length. Pyramid scoring acknowledges the potential variation in content selection across different acceptable summaries. However, this method is quite labor-intensive because it requires human assessors to perform detailed content analysis on the model summaries.
- CIDEr (Consensus-based Image Description Evaluation) [174]: is an evaluation metric primarily designed for assessing the quality of image captions in the context of image captioning tasks. It also gets used in text summarization to some extent. The fundamental idea behind CIDEr is that words

Table 3. A comparison of different Models on CNN/Daily Mail [125]

Models	Model	Rouge-1	Rouge-2	Rouge-L
Attentional RNN [125]	RNN	35.46	13.30	32.65
Pointer-Generator [161]	RNN	39.53	17.28	36.38
DynamicConv [186]	CNN	39.84	16.25	36.73
TaLK Convolution [98]	CNN	40.59	18.97	36.81
RL with intra-attention [143]	RL	41.16	15.75	39.08
RNN-ext+abs+RL+rerank [28]	RNN+RL	39.66	15.85	37.34
BILSTM+GNN+LSTM+POINTER [50]	GNN+LSTM	38.10	16.10	33.20
Graph-Based Attentional LSTM [168]	GNN+LSTM	38.10	13.90	34.00
Transformer [173]	Transformer	39.50	16.06	36.63
PEGASUS [196]	Transformer	44.17	21.47	41.11
BART [91]	Transformer	44.16	21.28	40.90
SEASON [180]	Transformer	46.38	22.83	43.18
BART.GPT-4 [104]	Transformer	63.22	44.70	-

that are more important to a description should have higher weights.

- BERTScore [198]: is an automatic evaluation metric for natural language generation tasks, including text summarization. Unlike traditional metrics like ROUGE and BLEU which rely on n-gram overlaps, BERTScore leverages the contextual embeddings from the pre-trained BERT model to evaluate the generated text.
- Moverscore [202]: is based on two fundamental principles: the use of contextualized embeddings and the Earth Mover's Distance (EMD), also known as the Wasserstein distance. Contextualized embeddings, such as BERT embeddings, represent words or phrases within the context they appear, providing a more meaningful representation of the text. The Earth Mover's Distance is a measure of the distance between two probability distributions over a region, and it's used here to measure the distance between the embeddings of the generated summary and the reference summary.

Table 3 shows some experimental results of popular models on CNN/Daily Mail. Although automatic metrics are widely used, they do not always align well with human judgments of summary quality. Human evaluation is considered the gold standard, but it's time-consuming and costly. Thus, a combination of automatic and human evaluation is often used in practice.

5. Summary

5.1. *Challenge and Future*

Text summarization is an intriguing and challenging task in the realm of natural language processing. In recent years, significant advancements have been made with the aid of deep learning (DL) models. Novel concepts such as neural embedding, at-

tention mechanism, self-attention, Transformer, BERT, and GPT-4 have propelled the field forward, resulting in rapid progress over the past decade. However, despite these advancements, there are still notable challenges that need to be addressed. This section aims to highlight some of the remaining challenges in text summarization and explore potential research directions that can contribute to further advancements in the field. By addressing these challenges and exploring new avenues, we can continue to push the boundaries of text summarization and unlock its full potential.

One critical aspect is the need to understand the context of the document, encompassing semantics, syntactic structure, and discourse organization. Deep learning models often struggle with complex or ambiguous language, idiomatic expressions, and domain-specific jargon, making it difficult to achieve accurate and meaningful summaries.

A well-crafted summary should exhibit coherence and cohesion, ensuring that ideas logically connect and the text flows smoothly. Models must generate summaries that preserve the integrity of the original text's meaning without introducing inconsistencies or redundancies. This requires a deep understanding of the main ideas, supporting details, and their interrelationships. Identifying important content poses a significant challenge, as it necessitates discerning the relevance and significance of various elements in the document.

Summarizing long documents, such as legal or research papers, presents additional hurdles. These documents often contain complex sentence structures, advanced vocabulary, and important information distributed throughout the text. Creating concise summaries that capture the key points while maintaining accuracy becomes a daunting task. Furthermore, the lack of labeled training data exacerbates the challenges. Supervised learning approaches for text summarization rely on substantial amounts of annotated data, which can be expensive and time-consuming to create.

Evaluating the quality of generated summaries is another ongoing challenge. Automatic evaluation metrics, such as ROUGE, BLEU, or BERTScore, do not always align perfectly with human judgment. Manual evaluation, while providing more accurate insights, is a labor-intensive process. Overcoming these challenges requires the development of better evaluation metrics that align more closely with human perceptions of summary quality.

Summarization tasks become more intricate when they are domain-specific, such as in medical or legal contexts. These domains often employ specialized language, requiring a higher level of understanding and accuracy. Additionally, summarizing information from multiple documents introduces further complexities. Models must eliminate redundant information, handle potentially conflicting details, and synthesize the most relevant content from various sources.

As the field progresses, advancements in deep learning models, particularly transformer-based architectures like BERT, GPT-4, and T5, offer promising opportunities for improved performance in text summarization tasks. Fine-tuning pre-

trained models on specific summarization objectives has shown great potential and is expected to continue. Furthermore, as we become increasingly interconnected globally, there will be a growing demand for models capable of summarizing text in different languages or even across languages.

Addressing the challenges of explainability, transparency, bias, data efficiency, multi-modal summarization, and personalized summarization are areas that will likely receive significant attention in future research. Explainable and transparent AI models are becoming increasingly important, and efforts to develop models that can provide reasoning for their decisions are expected. The development of better evaluation metrics, mitigating biases, exploring more data-efficient methods, handling multi-modal information, and catering to personalized summarization needs are all potential avenues for further advancement in the field.

5.2. *Conclusion*

This article presents a comprehensive survey of over 100 deep learning models developed in the last decade, highlighting their significant advancements in various text summarization tasks. Additionally, we provide an overview of popular summarization datasets and conduct a quantitative analysis to assess the performance of these models on several public benchmarks. Furthermore, we address open challenges in the field and propose potential future research directions.

References

[1] Deen Mohammad Abdullah and Yllias Chali. Towards generating query to perform query focused abstractive summarization using pre-trained model. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 80–85, Dublin, Ireland, December 2020. Association for Computational Linguistics.

[2] Laith Abualigah, Mohammad Qassem Bashabsheh, Hamzeh Alabool, and Mohammad Shehab. Text summarization: A brief review. *Recent Advances in NLP: The Case of Arabic Language*, 874:1, 2019.

[3] Stergos Afantenos, Vangelis Karkaletsis, and Panagiotis Stamatopoulos. Summarization from medical documents: a survey. *Artificial intelligence in medicine*, 33(2):157–177, 2005.

[4] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. Text summarization techniques: a brief survey. *arXiv preprint arXiv:1707.02268*, 2017.

[5] Narendra Andhale and Laxmi A Bewoor. An overview of text summarization techniques. In *2016 international conference on computing communication control and automation (ICCUBEA)*, pages 1–7. IEEE, 2016.

[6] Eiji Aramaki, Yasuhide Miura, Masatsugu Tonoike, Tomoko Ohkuma, Hiroshi Masuichi, and Kazuhiko Ohe. Text2table: Medical text summarization system based on named entity recognition and modality identification. In *Proceedings of the BioNLP 2009 Workshop*, pages 185–192, 2009.

[7] Aqil M Azmi and Suha Al-Thanyyan. A text summarizer for arabic. *Computer Speech & Language*, 26(4):260–273, 2012.

[8] Yu Bai, Yang Gao, and Heyan Huang. Cross-lingual abstractive summarization with

limited parallel resources. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6910–6924, Online, August 2021. Association for Computational Linguistics.

[9] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[10] Michele Banko, Vibhu O Mittal, and Michael J Witbrock. Headline generation based on statistical translation. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 318–325, 2000.

[11] Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *Preprint*, 2020.

[12] Regina Barzilay and Kathleen R McKeown. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328, 2005.

[13] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020.

[14] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000.

[15] Neelima Bhatia and Arunima Jaiswal. Automatic text summarization and it's methods-a review. In *2016 6th international conference-cloud system and big data engineering (Confluence)*, pages 65–72. IEEE, 2016.

[16] Florian Boudin, Stéphane Huet, and Juan-Manuel Torres-Moreno. A graph-based approach to cross-language multi-document summarization. *Polibits*, 43:113–118, 2011.

[17] Ronald Brandow, Karl Mitze, and Lisa F Rau. Automatic condensation of electronic publications by sentence selection. *Information Processing & Management*, 31(5):675–685, 1995.

[18] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[19] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[20] Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel Weld. TLDR: Extreme summarization of scientific documents. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4766–4777, Online, November 2020. Association for Computational Linguistics.

[21] Xiaoyan Cai and Wenjie Li. Mutually reinforced manifold-ranking based relevance propagation model for query-focused multi-document summarization. *IEEE transactions on audio, speech, and language processing*, 20(5):1597–1607, 2012.

[22] Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. Ranking with recursive neural networks and its application to multi-document summarization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2153–2159. AAAI Press, 2015.

[23] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information*

retrieval, pages 335–336, 1998.

[24] Yllias Chali and Sadid A. Hasan. On the effectiveness of using sentence compression models for query-focused multi-document summarization. In *Proceedings of COLING 2012*, pages 457–474, Mumbai, India, December 2012. The COLING 2012 Organizing Committee.

[25] Yllias Chali and Sadid A Hasan. Query-focused multi-document summarization: Automatic data annotations and supervised learning approaches. *Natural Language Engineering*, 18(1):109–145, 2012.

[26] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online, November 2020. Association for Computational Linguistics.

[27] Ping Chen and Rakesh Verma. A query-based medical information summarization system using ontology knowledge. In *19th IEEE Symposium on Computer-Based Medical Systems (CBMS'06)*, pages 37–42. IEEE, 2006.

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

[29] Yen-Chun Chen, Zhe Gan, Yu Cheng, Jingzhou Liu, and Jingjing Liu. Distilling knowledge learned in BERT for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7893–7905, Online, July 2020. Association for Computational Linguistics.

[30] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[31] Sumit Chopra, Michael Auli, and Alexander M. Rush. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California, June 2016. Association for Computational Linguistics.

[32] Hans Christian, Mikhael Pramodana Agus, and Derwin Suhartono. Single document automatic text summarization using term frequency-inverse document frequency (tf-idf). *ComTech: Computer, Mathematics and Engineering Applications*, 7(4):285–294, 2016.

[33] Freddy Chua and Sitaram Asur. Automatic summarization of events from social media. *Proceedings of the International AAAI Conference on Web and Social Media*, 7(1):81–90, Aug. 2021.

[34] Adam Coates, Brody Huval, Tao Wang, David Wu, Bryan Catanzaro, and Ng Andrew. Deep learning with cots hpc systems. In *International conference on machine learning*, pages 1337–1345. PMLR, 2013.

[35] Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[36] Document Understanding Conference. Duc 2004. `https://duc.nist.gov/duc2004/tasks.html`, 2004. Updated: 2011-03-24.

[37] Hal Daumé III and Daniel Marcu. Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 305–312, Sydney, Australia, July 2006. Association for Computational Linguistics.

[38] Daisy Deng. Bootstrap your text summarization solution with the latest release from nlp-recipes. `https://techcommunity.microsoft.com/t5/ai-customer-engineering-team/bootstrap-your-text-summarization-solution-with-the-latest/ba-p/1268809`, 2020. Updated: 2020-03-31.

[39] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[40] Xuan-Dung Doan, Le-Minh Nguyen, and Khac-Hoai Nam Bui. Multi graph neural network for extractive long document summarization. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5870–5875, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics.

[41] Luobing Dong, Meghana N Satpute, Weili Wu, and Ding-Zhu Du. Two-phase multi-document summarization through content-attention-based subtopic detection. *IEEE Transactions on Computational Social Systems*, 8(6):1379–1392, 2021.

[42] Bonnie Dorr, David Zajic, and Richard Schwartz. Hedge trimmer: A parse-and-trim approach to headline generation. In *Proceedings of the HLT-NAACL 03 Text Summarization Workshop*, pages 1–8, 2003.

[43] Xiangyu Duan, Mingming Yin, Min Zhang, Boxing Chen, and Weihua Luo. Zero-shot cross-lingual abstractive sentence summarization through teaching generation and attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3162–3172, Florence, Italy, July 2019. Association for Computational Linguistics.

[44] Harold P Edmundson. New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285, 1969.

[45] Gunes Erkan and Dragomir Radev. Lexpagerank: Prestige in multi-document text summarization. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 365–371, 2004.

[46] Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479, 2004.

[47] Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy, July 2019. Association for Computational Linguistics.

[48] James R Faeder, Michael L Blinov, and William S Hlavacek. Rule-based modeling of biochemical systems with bionetgen. *Systems biology*, pages 113–167, 2009.

[49] Xiachong Feng, Xiaocheng Feng, and Bing Qin. A survey on dialogue summarization: Recent advances and new frontiers. *arXiv preprint arXiv:2107.03175*, 2021.

[50] Patrick Fernandes, Miltiadis Allamanis, and Marc Brockschmidt. Structured neural summarization, 2021.

[51] Seeger Fisher and Brian Roark. Query-focused summarization by supervised sentence ranking and skewed word distributions. In *Proceedings of the Document Understanding Conference, DUC-2006, New York, USA*, 2006.

[52] Maria Fuentes, Enrique Alfonseca, and Horacio Rodríguez. Support vector machines for query-focused summarization trained and evaluated on pyramid data. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 57–60, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[53] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.

[54] Pascale Fung and Grace Ngai. One story, one flow: Hidden markov story models for multilingual multidocument summarization. *ACM Transactions on Speech and Language Processing (TSLP)*, 3(2):1–16, 2006.

[55] Filippo Galgani, Paul Compton, and Achim Hoffmann. Combining different summarization techniques for legal text. In *Proceedings of the workshop on innovative hybrid approaches to the processing of textual data*, pages 115–123, 2012.

[56] Mahak Gambhir and Vishal Gupta. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, 47:1–66, 2017.

[57] Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd international conference on computational linguistics (Coling 2010)*, pages 340–348, 2010.

[58] Saeedeh Gholamrezazadeh, Mohsen Amini Salehi, and Bahareh Gholamzadeh. A comprehensive survey on text summarization systems. In *2009 2nd International Conference on Computer Science and its Applications*, pages 1–6. IEEE, 2009.

[59] Jade Goldstein, Mark Kantrowitz, Vibhu Mittal, and Jaime Carbonell. Summarizing text documents: Sentence selection and evaluation metrics. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 121–128, 1999.

[60] Yihong Gong and Xin Liu. Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–25, 2001.

[61] Quentin Grail, Julien Perez, and Eric Gaussier. Globalizing bert-based transformer architectures for long document summarization. In *Proceedings of the 16th conference of the European chapter of the Association for Computational Linguistics: Main volume*, pages 1792–1810, 2021.

[62] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[63] Max Grusky, Mor Naaman, and Yoav Artzi. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[64] Wang Guan, Ivan Smetannikov, and Man Tianxing. Survey on automatic text summarization and transformer models applicability. In *Proceedings of the 2020 1st In-*

ternational Conference on Control, Robotics and Intelligent System, pages 176–184, 2020.

[65] Vishal Gupta. Hybrid algorithm for multilingual summarization of hindi and punjabi documents. In *Mining Intelligence and Knowledge Exploration: First International Conference, MIKE 2013, Tamil Nadu, India, December 18-20, 2013. Proceedings*, pages 717–727. Springer, 2013.

[66] Vishal Gupta and Gurpreet Singh Lehal. A survey of text summarization extractive techniques. *Journal of emerging technologies in web intelligence*, 2(3):258–268, 2010.

[67] Johan Hasselqvist, Niklas Helmertz, and Mikael Kågebäck. Query-based abstractive summarization using neural networks. *arXiv preprint arXiv:1712.06100*, 2017.

[68] Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377, San Diego, California, June 2016. Association for Computational Linguistics.

[69] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[70] Meishan Hu, Aixin Sun, and Ee-Peng Lim. Comments-oriented document summarization: understanding documents with readers' feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 291–298, 2008.

[71] Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online, June 2021. Association for Computational Linguistics.

[72] X Huang, A Acero, and H Hon. A guide to theory, algorithm, and system development. *Spoken Language Processing. Prentice-Hall*, 2001.

[73] Dongmin Hyun, Xiting Wang, Chayoung Park, Xing Xie, and Hwanjo Yu. Generating multiple-length summaries via reinforcement learning for unsupervised sentence summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2939–2951, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[74] Anubhav Jangra, Sourajit Mukherjee, Adam Jatowt, Sriparna Saha, and Mohammad Hasanuzzaman. A survey on multi-modal summarization. *ACM Computing Surveys*, 2021.

[75] Baoyu Jing, Zeyu You, Tao Yang, Wei Fan, and Hanghang Tong. Multiplex graph neural network for extractive text summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 133–139, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[76] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[77] Ambedkar Kanapala, Sukomal Pal, and Rajendra Pamula. Text summarization from legal documents: a survey. *Artificial Intelligence Review*, 51:371–402, 2019.

[78] Huan Yee Koh, Jiaxin Ju, Ming Liu, and Shirui Pan. An empirical survey on long document summarization: Datasets, models, and metrics. *ACM computing surveys*, 55(8):1–35, 2022.

[79] Anastassia Kornilova and Vladimir Eidelman. BillSum: A corpus for automatic summarization of US legislation. In *Proceedings of the 2nd Workshop on New Frontiers*

in Summarization, pages 48–56, Hong Kong, China, November 2019. Association for Computational Linguistics.

[80] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[81] Wojciech Kryściński, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. Booksum: A collection of datasets for long-form narrative summarization. *arXiv preprint arXiv:2105.08209*, 2021.

[82] Julian Kupiec, Jan Pedersen, and Francine Chen. A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 68–73, 1995.

[83] Moreno La Quatra and Luca Cagliero. End-to-end training for financial report summarization. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 118–123, Barcelona, Spain (Online), December 2020. COLING.

[84] Md Tahmid Rahman Laskar, Enamul Hoque, and Jimmy Huang. Query focused abstractive summarization via incorporating query relevance and transfer learning with transformer models. In *Advances in Artificial Intelligence: 33rd Canadian Conference on Artificial Intelligence, Canadian AI 2020, Ottawa, ON, Canada, May 13–15, 2020, Proceedings 33*, pages 342–348. Springer, 2020.

[85] Dawn Lawrie, W Bruce Croft, and Arnold Rosenberg. Finding topic words for hierarchical summarization. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 349–357, 2001.

[86] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[87] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[88] Chang-Shing Lee, Zhi-Wei Jian, and Lin-Kai Huang. A fuzzy ontology and its application to news summarization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(5):859–880, 2005.

[89] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 09 2019.

[90] Anton Leuski, Chin-Yew Lin, Liang Zhou, Ulrich Germann, Franz Josef Och, and Eduard Hovy. Cross-lingual c*st*rd: English access to hindi information. *ACM Transactions on Asian Language Information Processing*, 2(3):245–269, sep 2003.

[91] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics.

[92] Zhengpeng Li, Jiansheng Wu, Jiawei Miao, and Xinmiao Yu. News headline generation based on improved decoder from transformer. *Scientific Reports*, 12(1):11648, 2022.

[93] Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Win-

nie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. XGLUE: A new benchmark dataset for cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018, Online, November 2020. Association for Computational Linguistics.

[94] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[95] Chin-Yew Lin and Eduard Hovy. Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 workshop on automatic summarization*, pages 45–51, 2002.

[96] Hui Lin, Jeff Bilmes, and Shasha Xie. Graph-based submodular selection for extractive summarization. In *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 381–386. IEEE, 2009.

[97] Hui Lin and Vincent Ng. Abstractive summarization: A survey of the state of the art. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):9815–9822, Jul. 2019.

[98] Vasileios Lioutas and Yuhong Guo. Time-aware large kernel convolutions. In *International Conference on Machine Learning*, pages 6172–6183. PMLR, 2020.

[99] Linqing Liu, Yao Lu, Min Yang, Qiang Qu, Jia Zhu, and Hongyan Li. Generative adversarial network for abstractive text summarization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[100] Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*, 2018.

[101] Yang Liu. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*, 2019.

[102] Yang Liu and Mirella Lapata. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy, July 2019. Association for Computational Linguistics.

[103] Yike Liu, Tara Safavi, Abhilash Dighe, and Danai Koutra. Graph summarization methods and applications: A survey. *ACM computing surveys (CSUR)*, 51(3):1–34, 2018.

[104] Yixin Liu, Alexander R. Fabbri, Pengfei Liu, Dragomir Radev, and Arman Cohan. On learning to summarize with large language models as references, 2023.

[105] Yizhu Liu, Zhiyi Luo, and Kenny Zhu. Controlling length in abstractive summarization using a convolutional neural network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4110–4119, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[106] Hans Peter Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.

[107] Takuya Makino, Tomoya Iwakura, Hiroya Takamura, and Manabu Okumura. Global optimization under length constraint for neural text summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1039–1048, Florence, Italy, July 2019. Association for Computational Linguistics.

[108] Inderjeet Mani and Eric Bloedorn. Machine learning of generic and user-focused summarization. In *AAAI/IAAI*, pages 821–826, 1998.

[109] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit.

In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[110] Yuning Mao, Ming Zhong, and Jiawei Han. CiteSum: Citation text-guided scientific extreme summarization and domain adaptation with limited supervision. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10922–10935, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[111] Kathleen McKeown and Dragomir R Radev. Generating summaries of multiple news articles. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 74–82, 1995.

[112] Kathleen R McKeown, Regina Barzilay, David Evans, Vasileios Hatzivassiloglou, Judith L Klavans, Ani Nenkova, Carl Sable, Barry Schiffman, and Sergey Sigelman. Tracking and summarizing news on a daily basis with columbia's newsblaster. In *Proceedings of the human language technology conference*, pages 280–285. San Diego, CA, 2002.

[113] Rui Meng, Khushboo Thaker, Lei Zhang, Yue Dong, Xingdi Yuan, Tong Wang, and Daqing He. Bringing structure into summaries: a faceted summarization dataset for long scientific documents. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1080–1089, Online, August 2021. Association for Computational Linguistics.

[114] Alessio Micheli. Neural network for graphs: A contextual constructive approach. *IEEE Transactions on Neural Networks*, 20(3):498–511, 2009.

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

[116] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004.

[117] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[118] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[119] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[120] N Moratanch and S Chitrakala. A survey on extractive text summarization. In *2017 international conference on computer, communication and signal processing (ICCCSP)*, pages 1–6. IEEE, 2017.

[121] Gianluca Moro and Luca Ragazzi. Semantic self-segmentation for abstractive summarization of long documents in low-resource regimes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11085–11093, Jun. 2022.

[122] Mohammed Elsaid Moussa, Ensaf Hussein Mohamed, and Mohamed Hassan Haggag. A survey on opinion summarization techniques for social media. *Future Computing and Informatics Journal*, 3(1):82–109, 2018.

[123] Nikita Munot and Sharvari S Govilkar. Comparative study of text summarization methods. *International Journal of Computer Applications*, 102(12), 2014.

[124] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3075–3081. AAAI Press, 2017.

[125] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gùlçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany, August 2016. Association for Computational Linguistics.

[126] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

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

[128] Narges Nazari and MA Mahdavi. A survey on automatic text summarization. *Journal of AI and Data Mining*, 7(1):121–135, 2019.

[129] Preksha Nema, Mitesh M. Khapra, Anirban Laha, and Balaraman Ravindran. Diversity driven attention model for query-based abstractive summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1063–1072, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[130] Ani Nenkova and Kathleen McKeown. A survey of text summarization techniques. *Mining text data*, pages 43–76, 2012.

[131] Ani Nenkova and Rebecca Passonneau. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.

[132] Ani Nenkova and Lucy Vanderwende. The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005*, 101, 2005.

[133] OpenAI. Gpt-4 technical report, 2023.

[134] Constantin Orăsan and Oana Andreea Chiorean. Evaluation of a cross-lingual Romanian-English multi-document summariser. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA).

[135] Jessica Ouyang, Boya Song, and Kathy McKeown. A robust abstractive system for cross-lingual summarization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2025–2031, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[136] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

[137] You Ouyang, Wenjie Li, Sujian Li, and Qin Lu. Applying regression models to query-focused multi-document summarization. *Inf. Process. Manage.*, 47(2):227–237, mar 2011.

[138] Tatsuro Oya, Yashar Mehdad, Giuseppe Carenini, and Raymond Ng. A template-based abstractive meeting summarization: Leveraging summary and source text relationships. In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, pages 45–53, Philadelphia, Pennsylvania, U.S.A., June 2014. Association for Computational Linguistics.

[139] Makbule Gulcin Ozsoy, Ferda Nur Alpaslan, and Ilyas Cicekli. Text summarization using latent semantic analysis. *Journal of Information Science*, 37(4):405–417, 2011.

[140] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[141] Daraksha Parveen, Hans-Martin Ramsl, and Michael Strube. Topical coherence for graph-based extractive summarization. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1949–1954, 2015.

[142] Alkesh Patel, Tanveer Siddiqui, and US Tiwary. A language independent approach to multilingual text summarization. *Large scale semantic access to content (text, image, video, and sound)*, pages 123–132, 2007.

[143] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017.

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

[145] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[146] Seth Polsley, Pooja Jhunjhunwala, and Ruihong Huang. CaseSummarizer: A system for automated summarization of legal texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 258–262, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[147] Dragomir Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Çelebi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, Danyu Liu, Jahna Otterbacher, Hong Qi, Horacio Saggion, Simone Teufel, Michael Topper, Adam Winkel, and Zhu Zhang. MEAD - a platform for multidocument multilingual text summarization. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA).

[148] Dragomir R Radev, Sasha Blair-Goldensohn, Zhu Zhang, and Revathi Sundara Raghavan. Newsinessence: A system for domain-independent, real-time news clustering and multi-document summarization. In *Proceedings of the first international conference on Human language technology research*, pages 1–4, 2001.

[149] Dragomir R Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938, 2004.

[150] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training, 2018.

[151] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[152] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), jan 2020.

[153] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[154] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[155] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[156] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[157] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

[158] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[159] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[160] Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. MLSUM: The multilingual summarization corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8051–8067, Online, November 2020. Association for Computational Linguistics.

[161] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[162] Eva Sharma, Chen Li, and Lu Wang. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213, Florence, Italy, July 2019. Association for Computational Linguistics.

[163] Chao Shen and Tao Li. Learning to rank for query-focused multi-document summarization. In *2011 IEEE 11th International Conference on Data Mining*, pages 626–634. IEEE, 2011.

[164] Josef Steinberger, Karel Jezek, et al. Using latent semantic analysis in text summarization and summary evaluation. *Proc. ISIM*, 4(93-100):8, 2004.

[165] Milan Straka, Nikita Mediankin, Tom Kocmi, Zdeněk Žabokrtský, Vojtěch Hudeček, and Jan Hajič. SumeCzech: Large Czech news-based summarization dataset. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[166] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018.

[167] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[168] Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1171–1181, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[169] Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. From neural sentence summarization to headline generation: A coarse-to-fine approach. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 4109–4115. AAAI Press, 2017.

[170] Oguzhan Tas and Farzad Kiyani. A survey automatic text summarization. *PressAcademia Procedia*, 5(1):205–213, 2007.

[171] Khushboo S Thakkar, Rajiv V Dharaskar, and MB Chandak. Graph-based algorithms for text summarization. In *2010 3rd International Conference on Emerging Trends in Engineering and Technology*, pages 516–519. IEEE, 2010.

[172] Gian Lorenzo Thione, Martin van den Berg, Livia Polanyi, and Chris Culy. Hybrid text summarization: Combining external relevance measures with structural analysis. In *Text Summarization Branches Out*, pages 51–55, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[173] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[174] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[175] Jesse Vig, Alexander Fabbri, Wojciech Kryscinski, Chien-Sheng Wu, and Wenhao Liu. Exploring neural models for query-focused summarization. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1455–1468, Seattle, United States, July 2022. Association for Computational Linguistics.

[176] Xiaojun Wan. Using bilingual information for cross-language document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1546–1555, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[177] Xiaojun Wan, Huiying Li, and Jianguo Xiao. Cross-language document summarization based on machine translation quality prediction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 917–926, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[178] Xiaojun Wan and Jianwu Yang. Multi-document summarization using cluster-based link analysis. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 299–306, 2008.

[179] Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219, Online, July 2020. Association for Computational Linguistics.

[180] Fei Wang, Kaiqiang Song, Hongming Zhang, Lifeng Jin, Sangwoo Cho, Wenlin Yao,

Xiaoyang Wang, Muhao Chen, and Dong Yu. Salience allocation as guidance for abstractive summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6094–6106, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[181] Jiaan Wang, Fandong Meng, Duo Zheng, Yunlong Liang, Zhixu Li, Jianfeng Qu, and Jie Zhou. A survey on cross-lingual summarization. *Transactions of the Association for Computational Linguistics*, 10:1304–1323, 2022.

[182] Lu Wang and Claire Cardie. Domain-independent abstract generation for focused meeting summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1395–1405, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[183] Mark Wasson. Using leading text for news summaries: Evaluation results and implications for commercial summarization applications. In *COLING 1998 Volume 2: The 17th International Conference on Computational Linguistics*, 1998.

[184] Jonathan J Webster and Chunyu Kit. Tokenization as the initial phase in nlp. In *COLING 1992 volume 4: The 14th international conference on computational linguistics*, 1992.

[185] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement learning*, pages 5–32, 1992.

[186] Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. *arXiv preprint arXiv:1901.10430*, 2019.

[187] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265, 2018.

[188] Wen Xiao and Giuseppe Carenini. Extractive summarization of long documents by combining global and local context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3011–3021, Hong Kong, China, November 2019. Association for Computational Linguistics.

[189] Shusheng Xu, Xingxing Zhang, Yi Wu, and Furu Wei. Sequence level contrastive learning for text summarization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11556–11565, Jun. 2022.

[190] Divakar Yadav, Jalpa Desai, and Arun Kumar Yadav. Automatic text summarization methods: A comprehensive review. *arXiv preprint arXiv:2204.01849*, 2022.

[191] Jin-ge Yao, Xiaojun Wan, and Jianguo Xiao. Phrase-based compressive cross-language summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 118–127, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[192] Tiezheng Yu, Zihan Liu, and Pascale Fung. AdaptSum: Towards low-resource domain adaptation for abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5892–5904, Online, June 2021. Association for Computational Linguistics.

[193] Ruifeng Yuan, Zili Wang, Ziqiang Cao, and Wenjie Li. Few-shot query-focused summarization with prefix-merging. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3704–3714, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[194] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang,

et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

[195] Jiajun Zhang, Yu Zhou, and Chengqing Zong. Abstractive cross-language summarization via translation model enhanced predicate argument structure fusing. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(10):1842–1853, 2016.

[196] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR, 2020.

[197] Shiyue Zhang, Asli Celikyilmaz, Jianfeng Gao, and Mohit Bansal. EmailSum: Abstractive email thread summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6895–6909, Online, August 2021. Association for Computational Linguistics.

[198] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[199] Weiwei Zhang, Jackie Chi Kit Cheung, and Joel Oren. Generating character descriptions for automatic summarization of fiction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7476–7483, Jul. 2019.

[200] Xingxing Zhang, Furu Wei, and Ming Zhou. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069, Florence, Italy, July 2019. Association for Computational Linguistics.

[201] Xueying Zhang, Yunjiang Jiang, Yue Shang, Zhaomeng Cheng, Chi Zhang, Xiaochuan Fan, Yun Xiao, and Bo Long. Dsgpt: Domain-specific generative pre-training of transformers for text generation in e-commerce title and review summarization. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2146–2150, 2021.

[202] Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578, Hong Kong, China, November 2019. Association for Computational Linguistics.

[203] Xiaojuan Zhao and Jun Tang. Query-focused summarization based on genetic algorithm. In *2010 International Conference on Measuring Technology and Mechatronics Automation*, volume 2, pages 968–971. IEEE, 2010.

[204] Shaohui Zheng, Zhixu Li, Jiaan Wang, Jianfeng Qu, An Liu, Lei Zhao, and Zhigang Chen. Long-document cross-lingual summarization. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 1084–1092, 2023.

[205] Xin Zheng, Aixin Sun, Jing Li, and Karthik Muthuswamy. Subtopic-driven multi-document summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3153–3162, Hong Kong, China, November 2019. Association for Computational Linguistics.

[206] Liang Zhou and Eduard Hovy. Headline summarization at isi. In *Proceedings of the HLT-NAACL 2003 text summarization workshop and document understanding*

conference (DUC 2003), pages 174–178. Citeseer, 2003.

[207] Liang Zhou and Eduard Hovy. Template-filtered headline summarization. In *Text Summarization Branches Out*, pages 56–60, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[208] Li Zhuang, Feng Jing, and Xiao-Yan Zhu. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 43–50, 2006.