Multi-document Summarization via Deep Learning Techniques: A Survey

CONGBO MA, WEI EMMA ZHANG, MINGYU GUO, and HU WANG,
The University of Adelaide
QUAN Z. SHENG, Macquarie University

Multi-document summarization (MDS) is an effective tool for information aggregation that generates an informative and concise summary from a cluster of topic-related documents. Our survey, the first of its kind, systematically overviews the recent deep-learning-based MDS models. We propose a novel taxonomy to summarize the design strategies of neural networks and conduct a comprehensive summary of the state of the art. We highlight the differences between various objective functions that are rarely discussed in the existing literature. Finally, we propose several future directions pertaining to this new and exciting field.

CCS Concepts: • **Computing methodologies** → **Natural language processing**; **Machine learning algorithms**; **Information extraction;**

Additional Key Words and Phrases: Multi-document summarization, deep neural networks, machine learning

ACM Reference format:
Congbo Ma, Wei Emma Zhang, Mingyu Guo, Hu Wang, and Quan Z. Sheng. 2022. Multi-document Summarization via Deep Learning Techniques: A Survey. *ACM Comput. Surv.* 55, 5, Article 102 (December 2022), 37 pages.
https://doi.org/10.1145/3529754

1 INTRODUCTION

In this era of rapidly advancing technology, the exponential increase of data availability makes analyzing and understanding text files a tedious, labor-intensive, and time-consuming task [65, 120]. The need to process this abundance of text data rapidly and efficiently calls for new, effective text summarization techniques. Text summarization is a key **natural language processing (NLP)** task that automatically converts a text, or a collection of texts within the same topic, into a concise summary that contains key semantic information that can be beneficial for many downstream applications such as creating news digests, search engines, and report generation [127].

Text can be summarized from one or several documents, resulting in **single-document summarization (SDS)** and **multi-document summarization (MDS)**. While simpler to perform, SDS may not produce comprehensive summaries because it does not make good use of related, or more recent, documents. Conversely, MDS generates more comprehensive and accurate summaries from

Authors' addresses: C. Ma, W. E. Zhang, M. Guo, and H. Wang, The University of Adelaide; emails: {congbo.ma, wei.e.zhang, mingyu.guo, hu.wang}@adelaide.edu.au; Q. Z. Sheng, Macquarie University; email: michael.sheng@mq.edu.au.

0360-0300/2022/12-ART102 $15.00
https://doi.org/10.1145/3529754

documents written at different times, covering different perspectives, but is accordingly more complicated as it tries to resolve potentially diverse and redundant information [151]. In addition, incredibly lengthy input documents often lead to model degradation [74]. It is challenging for models to retain the most critical contents of complex input sequences while generating a coherent, non-redundant, factually consistent, and grammatically readable summary. Therefore, MDS requires models to have stronger capabilities for analyzing the input documents, identifying and merging consistent information.

MDS enjoys a wide range of real-world applications, including summarization of news [44], scientific publications [172], emails [23, 176], product reviews [50], medical documents [1], lecture feedback [102], software project activities [3], and Wikipedia articles [97]. Recently, MDS technology has also received a great amount of industry attention; an intelligent multilingual news reporter bot named Xiaomingbot [166] was developed for news generation, which can summarize multiple news sources into one article and translate it into multiple languages. Massive application requirements and rapidly growing online data have promoted the development of MDS. Existing methods using traditional algorithms are based on **term frequency-inverse document frequency (TF-IDF)** [11, 131], clustering [52, 159], graphs [104, 158], and latent semantic analysis [8, 60]. Most of these works still generate summaries with manually crafted features [108, 158], such as sentence position features [12, 41], sentence length features [41], proper noun features [157], cue-phrase features [59], biased word features, sentence-to-sentence cohesion, and sentence-to-centroid cohesion.

Deep learning has gained enormous attention in recent years due to its success in various domains, for instance, computer vision [81], natural language processing [36], and multi-modal learning [67]. Both industry and academia have embraced deep learning to solve complex tasks due to its capability of capturing highly nonlinear relations of data. Moreover, deep-learning-based models reduce dependence on manual feature extraction and pre-knowledge in the field of linguistics, drastically improving the ease of engineering [152]. Therefore, deep-learning-based methods demonstrate outstanding performance in MDS tasks in most cases [21, 85, 94, 98, 101]. With recent dramatic improvements in computational power and the release of increasing numbers of public datasets, neural networks with deeper layers and more complex structures have been applied in MDS [93, 98], accelerating the development of text summarization with more powerful and robust models. These tasks are attracting attention in the natural language processing community; the number of research publications on deep-learning-based MDS has increased rapidly over the last 5 years. The prosperity of deep learning for summarization in both academia and industry requires a comprehensive review of current publications for researchers to better understand the process and research progress. However, most of the existing summarization survey papers are based on traditional algorithms instead of deep-learning-based methods or target general text summarization [39, 46, 61, 116, 142]. We have therefore surveyed recent publications on deep learning methods for MDS that, to the best of our knowledge, is the first comprehensive survey of this field. This survey has been designed to classify neural-based MDS techniques into diverse categories thoroughly and systematically. We also conduct a detailed discussion on the categorization and progress of these approaches to establish a clearer concept standing in the shoes of readers. We hope this survey provides a panoramic view for researchers, practitioners, and educators to quickly understand and step into the field of deep-learning-based MDS. The key contributions of this survey are threefold:

- We propose a categorization scheme to organize current research and provide a comprehensive review for deep-learning-based MDS techniques, including deep-learning-based models, objective functions, benchmark datasets, and evaluation metrics.

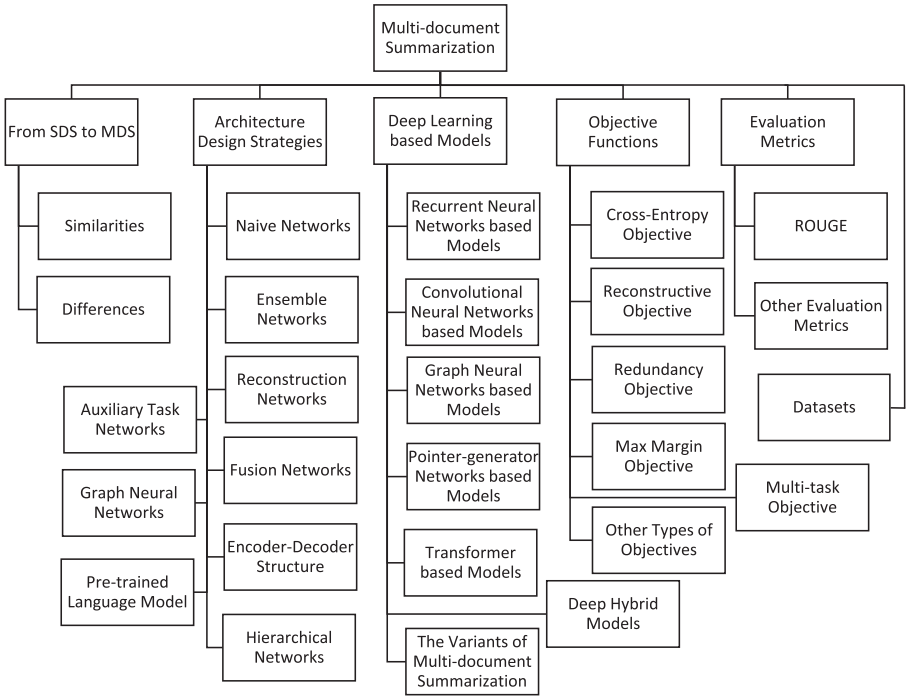

Fig. 1. Hierarchical structure of this survey.

- We review development movements and provide a systematic overview and summary of the state of the art. We also summarize nine network design strategies based on our extensive studies of the current models.
- We discuss the open issues of deep-learning-based multi-document summarization and identify the future research directions of this field. We also propose potential solutions for some discussed research directions.

Paper Selection. We used Google Scholar as the main search engine to select representative works from 2015 to 2021. High-quality papers were selected from top NLP and AI journals and conferences, including ACL,[1] EMNLP,[2] COLING,[3] NAACL,[4] AAAI,[5] ICML,[6] ICLR,[7] and IJCAI.[8] The major keywords we used include *multi-documentation summarization*, *summarization*, *extractive summarization*, *abstractive summarization*, *deep learning*, and *neural networks*.

Organization of the Survey. This survey will cover various aspects of recent advanced deep-learning-based works in MDS. Our proposed taxonomy categorizes the works from six aspects (Figure 1). To be more self-contained, in Section 2, we give the problem definition and the processing framework of text summarization and discuss similarities and differences between SDS and

[1]Annual Meeting of the Association for Computational Linguistics.
[2]Empirical Methods in Natural Language Processing.
[3]International Conference on Computational Linguistics
[4]Annual Conference of the North American Chapter of the Association for Computational Linguistics.
[5]AAAI Conference on Artificial Intelligence.
[6]International Conference on Machine Learning.
[7]International Conference on Learning Representations
[8]International Joint Conference on Artificial Intelligence.

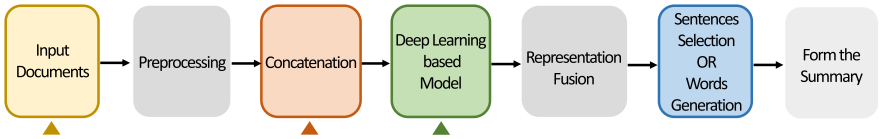

Fig. 2. The processing framework of text summarization. Each of the highlighted steps (the one with the triangle mark) indicates the differences between SDS and MDS.

MDS. Nine deep learning architecture design strategies, six deep-learning-based methods, and the variant tasks of MDS are presented in Section 3. Section 4 summarizes objective functions that guide the model optimization process in the reviewed literature, while evaluation metrics in Section 5 help readers choose suitable indices to evaluate the effectiveness of a model. Section 6 summarizes standard and variant MDS datasets. Finally, Section 7 discusses future research directions for deep-learning-based MDS, followed by conclusions in Section 8.

2 FROM SINGLE- TO MULTI-DOCUMENT SUMMARIZATION

Before we dive into the details of existing deep-learning-based techniques, we start by defining SDS and MDS and introducing the concepts used in both methods. The aim of MDS is to generate a concise and informative summary *Sum* from a collection of documents D. D denotes a cluster of topic-related documents $\{d_i \mid i \in [1, N]\}$, where N is the number of documents. Each document d_i consists of M_{d_i} sentences $\left\{s_{i,j} \mid j \in [1, M_{d_i}]\right\}$. $s_{i,j}$ refers to the jth sentence in the ith document. The standard summary Ref is called the *golden summary* or *reference summary*. Currently, most golden summaries are written by experts. We keep this notation consistent throughout the article.

To give readers a clear understanding of the processing of deep-learning-based summarization tasks, we summarize and illustrate the processing framework as shown in Figure 2. The first step is preprocessing input document(s), such as segmenting sentences, tokenizing non-alphabetic characters, and removing punctuation [144]. MDS models in particular need to select suitable concatenation methods to capture cross-document relations. Then, an appropriate deep-learning-based model is chosen to generate semantic-rich representation for downstream tasks. The next step is to fuse these various types of representation for later sentence selection or summary generation. Finally, document(s) are transformed into a concise and informative summary. Each of the highlighted steps in Figure 2 (indicated by triangles) indicates a difference between SDS and MDS. Based on this process, the research questions of MDS can be summarized as follows:

- How to capture the cross-document relations and in-document relations from the input documents?
- Compared to SDS, how to extract or generate salient information in a larger search space containing conflict, duplication, and complementary information?
- How to best fuse various representations from deep-learning-based models and external knowledge?
- How to comprehensively evaluate the performance of MDS models?

The following sections provide a comprehensive analysis of the similarities and differences between SDS and MDS.

2.1 Similarities between SDS and MDS

Existing SDS and MDS methods share the summarization construction types, learning strategies, evaluation indexes, and objective functions. SDS and MDS both seek to compress the document(s) into a short and informative summary. Existing summarization methods can be grouped into

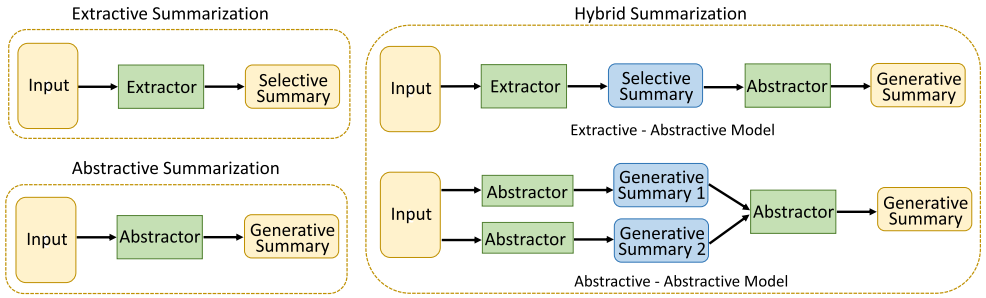

Fig. 3. Summarization construction types for text summarization.

abstractive summarization, *extractive summarization,* and *hybrid summarization* (Figure 3). Extractive summarization methods select salient snippets from the source documents to create informative summaries and generally contain two major components: *sentence ranking* and *sentence selection* [20, 112]. Abstractive summarization methods aim to present the main information of input documents by automatically generating summaries that are both succinct and coherent; this cluster of methods allows models to generate new words and sentences from a corpus pool [127]. Hybrid models are proposed to combine the advantages of both extractive and abstractive methods to process the input texts. Research on summarization focuses on two learning strategies. One strategy seeks to enhance the generalization performance by improving the architecture design of the end-to-end models [31, 44, 74, 98]. The other leverages external knowledge or other auxiliary tasks to complement summary selection or generation [19, 94]. Furthermore, both SDS and MDS aim to minimize the distance between machine-generated summary and golden summary. Therefore, SDS and MDS could share some indices to evaluate the performance of summarization models such as **Recall-Oriented Understudy for Gisting Evaluation (ROUGE**, see Section 5) and objective functions (see Section 4) to guide model optimization.

2.2 Differences between SDS and MDS

In the early stages of MDS, researchers directly applied SDS models to MDS [105]. However, a number of aspects in MDS are different from SDS, and these differences are also the breakthrough point for exploring the MDS models. We summarize the differences in the following five aspects:

- More diverse input document types
- Insufficient methods to capture cross-document relations
- High redundancy and contradiction across input documents
- Larger searching space but lack of sufficient training data
- Lack of evaluation metrics specifically designed for MDS

A defining different character between SDS and MDS is the number of input documents. MDS tasks deal with multiple sources of types that can be roughly divided into three groups:

- Many short sources, where each document is relatively short but the quantity of the input data is large. A typical example is product review summarization that aims to generate a short, informative summary from numerous individual reviews [5].
- Few long sources, for example, generating a summary from a group of news articles [44] or constructing a Wikipedia-style article from several web articles [97].
- Hybrid sources containing one or few long documents with several to many shorter documents, for example, news article(s) with several readers' comments to this news [92] or a scientific summary from a long paper with several short corresponding citations [172].

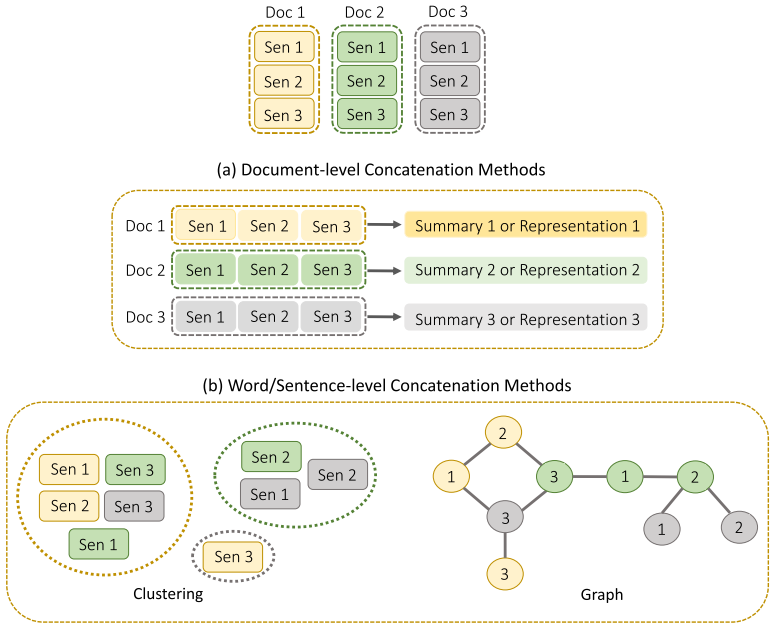

Fig. 4. The methods of hierarchical concatenation.

As SDS only uses one input document, no additional processing is required to assess relationships between SDS inputs. By their very nature, the multiple input documents used in MDS are likely to contain more contradictory, redundant, and complementary information [130]. MDS models therefore require sophisticated algorithms to identify and cope with redundancy and contradictions across documents to ensure that the final summary is comprehensive. Detecting these relations across documents can bring benefits for MDS models. In MDS tasks, there are two common methods to concatenate multiple input documents:

- Flat concatenation is a simple yet powerful concatenation method, where all input documents are spanned and processed as a flat sequence; to a certain extent, this method converts MDS to SDS tasks. Inputting flat-concatenated documents requires models to have a strong ability to process long sequences.
- Hierarchical concatenation is able to preserve cross-document relations. However, many existing deep learning methods do not make full use of this hierarchical relationship [44, 97, 160]. Taking advantage of hierarchical relations among documents instead of simply flat concatenating articles facilitates the MDS model to obtain representation with built-in hierarchical information, which in turn improves the effectiveness of the models. The input documents within a cluster describe a similar topic logically and semantically. Figure 4 illustrates two representative methods of hierarchical concatenation. Existing hierarchical concatenation methods either perform document-level condensing in a cluster separately [4] or process documents in word/sentence level inside a document cluster [6, 114, 160]. In Figure 4(a), the extractive or abstractive summaries or representations from the input documents are fused in the subsequent processes for final summary generation. The models using document-level concatenation methods are usually two-stage models. In Figure 4(b), sentences in the documents can be replaced by words. For word- or sentence-level concatenation methods, clustering algorithms and graph-based techniques are the most commonly

used methods. Clustering methods could help MDS models decrease redundancy and increase the information coverage for the generated summaries [114]. Sentence relation graph is able to model hierarchical relations among multi-documents as well [6, 172, 173]. Most of the graph construction methods utilize sentences as vertexes, and the edge between two sentences indicates their sentence-level relations [6]. Cosine similarity graph [41], discourse graph [30, 98, 173], semantic graph [126], and heterogeneous graph [160] can be used for building sentence graph structures. These graph structures could all serve as an external knowledge to improve the performance of MDS models.

In addition to capturing cross-document relations, hybrid summarization models can also be used to capture complex documents semantically, as well as to fuse disparate features that are more commonly adopted by MDS tasks. These models usually process data in two stages: extractive-abstractive and abstractive-abstractive (the right part of Figure 3). The two-stage models try to gather important information from source documents with extractive or abstractive methods at the first stage, to significantly reduce the length of documents. In the second stage, the processed texts are fed into an abstractive model to form final summaries [4, 85, 94, 97, 98].

Furthermore, conflict, duplication, and complementarity among multiple source documents require MDS models to have stronger abilities to handle complex information. However, applying the SDS model directly on MDS tasks is difficult to handle because of much higher redundancy [105]. Therefore, the MDS models are required to generate not only a coherent and complete summary but also more sophisticated algorithms to identify and cope with redundancy and contradictions across documents, ensuring that the final summary should be complete in itself. MDS also involves larger searching spaces but has smaller-scale training data than SDS, which sets obstacles for deep-learning-based models to learn adequate representation [105]. In addition, there are no specific evaluation metrics designed for MDS; however, existing SDS evaluation metrics cannot evaluate the relationship between the generated abstract and different input documents well.

3 DEEP-LEARNING-BASED MULTI-DOCUMENT SUMMARIZATION METHODS

Deep neural network (DNN) models learn multiple levels of representation and abstraction from input data and can fit data in a variety of research fields, such as computer vision [81] and natural language processing [36]. Deep learning algorithms replace manual feature engineering by learning distinctive features through back-propagation to minimize a given objective function. It is well known that linear solvable problems possess many advantages, such as being easily solved and having numerous theoretically proven supports; however, many NLP tasks are highly nonlinear. As theoretically proven by Hornik et al. [64], neural networks can fit any given continuous function as a universal approximator. For MDS tasks, DNNs also perform considerably better than traditional methods to effectively process large-scale documents and distill informative summaries due to their strong fitting abilities. In this section, we first introduce our novel taxonomy that generalizes nine neural network design strategies (Section 3.1). We then present the state-of-the-art DNN-based MDS models according to the main neural network architecture they adopt (Sections 3.2–3.7), before finishing with a brief introduction to MDS variant tasks (Section 3.8).

3.1 Architecture Design Strategies

Architecture design strategies play a critical role in deep-learning-based models, and many architectures have been applied to variant MDS tasks. Here, we have generalized the network architectures and summarized them into nine types based on how they generate or fuse semantic-rich and syntactic-rich representation to improve MDS model performance (Figure 5); these different architectures can also be used as basic structures or stacked on each other to obtain more diverse design

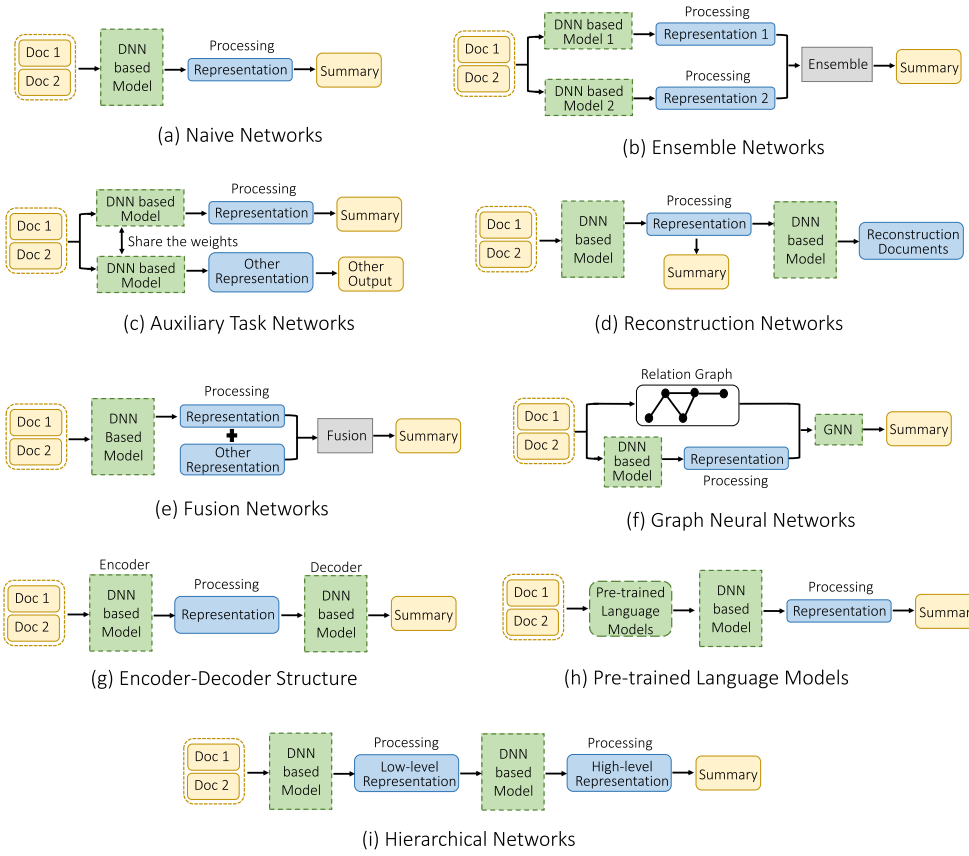

Fig. 5. Network design strategies.

strategies. In Figure 5, deep neural models are in green boxes and can be flexibly substituted with other backbone networks. The blue boxes indicate the neural embeddings processed by neural networks or heuristic-designed approaches, e.g., "sentence/document" or "other" representation. The explanation for each sub-figure is listed as follows:

- *Naive Networks (Figure 5(a)).* Multiple concatenated documents are input through DNN-based models to extract features. Word-level, sentence-level, or document-level representation is used to generate the downstream summary or select sentences. Naive networks represent the most naive model that lays the foundation for other strategies.
- *Ensemble Networks (Figure 5(b)).* Ensemble-based methods leverage multiple learning algorithms to obtain better performance than individual algorithms. To capture semantic-rich and syntactic-rich representation, ensemble networks feed input documents to multiple paths with different network structures or operations. Later on, the representation from different networks is fused to enhance model expression capability. The majority vote or the average score can be used to determine the final output.
- *Auxiliary Task Networks (Figure 5(c))* employ different tasks in the summarization models, where text classification, text reconstruction, or other auxiliary tasks serve as complementary representation learners to obtain advanced features. Meanwhile, auxiliary task networks also provide researchers with a solution to use appropriate data from

other tasks. In this strategy, parameter sharing schemes are used for jointly optimizing different tasks.

- *Reconstruction Networks (Figure 5(d))* optimize models from an unsupervised learning paradigm, which allows summarization models to overcome the limitation of insufficient annotated golden summaries. The use of such a paradigm enables generated summaries to be constrained in the natural language domain in a good manner.
- *Fusion Networks (Figure 5(e))* fuse representations generated from neural networks and hand-crafted features. These hand-crafted features contain adequate prior knowledge that facilitates the optimization of summarization models.
- *Graph Neural Networks (Figure 5(f)).* This strategy captures cross-document relations, crucial and beneficial for multi-document model training, by constructing graph structures based on the source documents, including word-, sentence-, or document-level information.
- *Encoder-Decoder Structure (Figure 5(g)).* The encoder embeds source documents into the hidden representation, i.e., word, sentence and document representation. This representation, containing compressed semantic and syntactic information, is passed to the decoder, which processes the latent embeddings to synthesize local and global semantic/syntactic information to produce the final summaries.
- *Pre-trained Language Models (Figure 5(h))* obtain contextualized text representation by predicting words or phrases based on their context using large amounts of the corpus, which can be further fine-tuned for downstream task adaption [37]. The models can fine-tune with randomly initialized decoders in an end-to-end fashion since transfer learning can assist the model training process [94].
- *Hierarchical Networks (Figure 5(i)).* Multiple documents are concatenated as inputs to feed into the first DNN-based model to capture low-level representation. Another DNN-based model is cascaded to generate high-level representation based on the previous ones. The hierarchical networks empower the model with the ability to capture abstract-level and semantic-level features more efficiently.

3.2 Recurrent-neural-network-based Models

Recurrent Neural Networks (RNNs) [137] excel in modeling sequential data by capturing sequential relations and syntactic/semantic information from word sequences. In RNN models, neurons are connected through hidden layers, and unlike other neural network structures, the inputs of each RNN neuron come not only from the word or sentence embedding but also from the output of the previous hidden state. Despite being powerful, vanilla RNN models often encounter gradient explosion or vanishing issues, so a large number of RNN-variants have been proposed. The most prevalent ones are **Long Short-Term Memory (LSTM)** [63], **Gated Recurrent Unit (GRU)** [32], and **Bi-directional Long Short-Term Memory (Bi-LSTM)** [66]. The DNN-based model in Figure 5 can be replaced with RNN-based models to design models.

RNN-based models have been used in MDS tasks since 2015. Cao et al. [20] proposed an RNN-based model termed *Ranking framework upon Recursive Neural Networks (R2N2)*, which leverages manually extracted words and sentence-level features as inputs. This model transfers the sentence ranking task into a hierarchical regression process, which measures the importance of sentences and constituents in the parsing tree. Zheng et al. [187] used a hierarchical RNN structure to utilize the subtopic information by extracting not only sentence and document embeddings but also topic embeddings. In this **SubTopic-Driven Summarization *(STDS)*** model, the readers' comments are seen as auxiliary documents and the model employs soft clustering to incorporate comment and sentence representation for further obtaining subtopic representation. Bražinskas et al. [16] introduced a GRU-based encoder-decoder architecture to minimize the diversity of opinions

reflecting the dominant views while generating multi-review summaries. Mao et al. [105] proposed a **maximal margin relevance guided reinforcement learning framework (RL-MMR)** to incorporate the advantages of neural sequence learning and statistical measures. The proposed soft attention for learning adequate representation allows more exploration of search space.

To leverage the advantage of the hybrid summarization model, Amplayo and Lapata [4] proposed a two-stage framework, viewing opinion summarization as an instance of multi-source transduction to distill salient information from source documents. The first stage of the model leverages a Bi-LSTM auto-encoder to learn word- and document-level representation; the second stage fuses multi-source representation and generates an opinion summary with a simple LSTM decoder combined with a vanilla attention mechanism [9] and a copy mechanism [156].

Since paired MDS datasets are rare and hard to obtain, Li et al. [93] developed an RNN-based framework to extract salient information vectors from sentences in input documents in an unsupervised manner. Cascaded attention retains the most relevant embeddings to reconstruct the original input sentence vectors. During the reconstruction process, the proposed model leverages a sparsity constraint to penalize trivial information in the output vectors. Also, Chu and Liu [31] proposed an unsupervised end-to-end abstractive summarization architecture called *MeanSum*. This LSTM-based model formalizes the product or business review summarization problem into two individual closed loops. Inspired by MeanSum, Coavoux et al. [33] used a two-layer standard LSTM to construct sentence representation for aspect-based multi-document abstractive summarization and discovered that the clustering strategy empowers the model to reward review diversity and handle contradictory ones.

3.3 Convolutional-neural-network-based Models

Convolutional neural networks (CNNs) [87] achieve excellent results in computer vision tasks. The convolution operation scans through the word/sentence embeddings and uses convolution kernels to extract important information from input data objects. Using a pooling operation at intervals can return simple to complex feature levels. CNNs have been proven to be effective for various NLP tasks in recent years [38, 77] as they can process natural language after sentence/word vectorization. Most of the CNN-based MDS models use CNNs for semantic and syntactic feature representation. As with RNN, CNN-based models can also replace DNN-based models in network design strategies (refer to Figure 5).

A simple way to use CNNs in MDS is by sliding multiple filters with different window sizes over the input documents for semantic representation. Cao et al. [21] proposed a hybrid CNN-based model *PriorSum* to capture latent document representation. The proposed representation learner slides over the input documents with filters of different window widths and two-layer max-over-time pooling operations [34] to fetch document-independent features that are more informative than using standard CNNs. Similarly, *HNet* [145] uses distinct CNN filters and max-over-time-pooling to generate salient feature representation for downstream processes. Cho et al. [29] also used different filter sizes in the *DPP-combined* model to extract low-level features. Yin and Pei [174] presented an unsupervised CNN-based model termed ***Novel Neural Language Model (NNLM)*** to extract sentence representation and diminish the redundancy of sentence selection. The NNLM framework contains only one convolution layer and one max-pooling layer, and both element-wise averaging sentence representation and context words representation are used to predict the next word. For aspect-based opinion summarization, Angelidis and Lapata [5] leveraged a CNN-based model to encode the product reviews that contain a set of segments for opinion polarity.

People with different background knowledge and understanding can produce different summaries of the same documents. To account for this variability, Zhang et al. [184] suggested an *MV-CNN* model that ensembles three individual models to incorporate multi-view learning and

Table 1. Multi-document Summarization Models Based on Graph Neural Networks

Models	Nodes	Edges	Edge Weights	GNN Methods
HeterDoc-SumGraph [160]	word, sentence, document	word-sentence, word-document	TF-IDF	Graph Attention Networks
Graph-based Neural MDS [173]	sentence	sentence-sentence	Personalized Discourse Graph	Graph Convolutional Networks
SemSentSum [6]	sentence	sentence-sentence	Cosine Similarity Graph Edge Removal Method	Graph Convolutional Networks
ScisummNet [172]	sentence	sentence-sentence	Cosine Similarity Graph	Graph Convolutional Networks

CNNs to improve the performance of MDS. In this work, three CNNs with dual-convolutional layers used multiple filters with different window sizes to extract distinct saliency scores of sentences.

To overcome the MDS bottlenecks of insufficient training data, Cao et al. [19] developed a *TCSum* model incorporating an auxiliary text classification sub-task into MDS to introduce more supervision signals. The text classification model uses a CNN descriptor to project documents onto the distributed representation and to classify input documents into different categories. The summarization model shares the projected sentence embedding from the classification model, and the TCSum model then chooses the corresponding category-based transformation matrices according to classification results to transform the sentence embedding into the summary embedding.

Unlike RNNs that support the processing of long time-serial signals, a naive CNN layer struggles to capture long-distance relations while processing sequential data due to the limitation of the fixed-sized convolutional kernels, each of which has a specific receptive field size. Nevertheless, CNN-based models can increase their receptive fields through formation of hierarchical structures to calculate sequential data in a parallel manner. Because of this highly parallelizable characteristic, training of CNN-based summarization models is more efficient than for RNN-based models. However, summarizing lengthy input articles is still a challenging task for CNN-based models because they are not skilled in modeling non-local relationships.

3.4 Graph-neural-network-based Models

CNNs have been successfully applied to many computer vision tasks to extract distinguished image features from the Euclidean space but struggle when processing non-Euclidean data. Natural language data consist of vocabularies and phrases with strong relations, which can be better represented with graphs than with sequential orders. **Graph neural networks (GNNs**, Figure 5(f)) are composed of an ideal architecture for NLP since they can model strong relations between entities semantically and syntactically. **Graph convolution networks (GCNs)** and **graph attention networks (GANs)** are the most commonly adopted GNNs because of their efficiency and simplicity for integration with other neural networks. These models first build a relation graph based on input documents, where nodes can be words, sentences, or documents, and edges capture the similarity among them. At the same time, input documents are fed into a DNN-based model to generate embeddings at different levels. The GNNs are then built over the top to capture salient contextual information. Table 1 describes the current GNN-based models used for MDS with details of nodes, edges, edge weights, and applied GNN methods.

Yasunaga et al. [173] developed a GCN-based extractive model to capture the relations between sentences. This model first builds a sentence-based graph and then feeds the pre-processed data into a GCN [78] to capture sentence-wise related features. Defined by the model, each sentence is regarded as a node and the relation between each pair of sentences is defined as an edge. Inside each document cluster, the sentence relation graph can be generated through a cosine similarity

graph [41], an approximate discourse graph [30], and the proposed personalized discourse graph. Both the sentence relation graph and sentence embeddings extracted by a sentence-level RNN are fed into GCN to produce the final sentence representation. With the help of a document-level GRU, the model generates cluster embeddings to fully aggregate features between sentences.

Similarly, Antognini and Faltings [6] proposed a GCN-based model named *SemSentSum* that constructs a graph based on sentence relations. In contrast to Yasunaga et al. [173], this work leverages external universal embeddings, pre-trained on the unrelated corpus, to construct a sentence semantic relation graph. Additionally, an edge removal method has been applied to deal with the sparse graph problems emphasizing high sentence similarities; if the weight of the edge is lower than a given threshold, the edge is removed. The sentence relation graph and sentence embeddings are fed into a GCN [78] to generate saliency estimation for extractive summaries.

Yasunaga et al. [172] also designed a GCN-based model for summarizing scientific papers. The proposed *ScisummNet* model uses not only the abstract of source scientific papers but also the relevant text from papers that cite the original source. The total number of citations is also incorporated into the model as an authority feature. A cosine similarity graph is applied to form the sentence relation graph, and GCNs are adopted to predict the sentence salience estimation from the sentence relation graph, authority scores, and sentence embeddings.

Existing GNN-based models focused mainly on the relationships between sentences and do not fully consider the relationships between words, sentences, and documents. To fill this gap, Wang et al. [160] proposed a heterogeneous GAN-based model, called *HeterDoc-SUM Graph*, that is specific for extractive MDS. This heterogeneous graph structure includes word, sentence, and document nodes, where sentence nodes and document nodes are connected according to the contained word nodes. Word nodes thus act as an intermediate bridge to connect the sentence and document nodes, and are used to better establish document-document, sentence-sentence, and sentence-document relations. TF-IDF values are used to weight word-sentence and word-document edges, and the node representation of these three levels is passed into the graph attention networks for model update. In each iteration, bi-directional updating of both word-sentence and word-document relations is performed to better aggregate cross-level semantic knowledge.

3.5 Pointer-generator-network-based Models

Pointer-generator (PG) networks [139] are proposed to overcome the problems of factual errors and high redundancy in the summarization tasks. This network has been inspired by Pointer Network [156], CopyNet [57], forced-attention sentence compression [107], and coverage mechanism from machine translation [153]. PG networks combine the **sequence-to-sequence (Seq2Seq)** model and pointer networks to obtain a united probability distribution allowing vocabularies to be selected from source texts or generated by machines. Additionally, the coverage mechanism prevents PG networks from consistently choosing the same phrases.

The ***Maximal Marginal Relevance (MMR)*** method is designed to select a set of salient sentences from source documents by considering both *importance* and *redundancy* indices [22]. The redundancy score controls sentence selection to minimize overlap with the existing summary. The MMR model adds a new sentence to the objective summary based on importance and redundancy scores until the summary length reaches a certain threshold. Inspired by MMR, Fabbri et al. [44] proposed an end-to-end ***Hierarchical MMR-Attention Pointer-generator (Hi-MAP)*** model to incorporate PG networks and MMR [22] for abstractive MDS. The Hi-MAP model improves PG networks by modifying attention weights (multiplying MMR scores by the original attention weights) to include better important sentences in, and filter redundant information from, the summary. Similarly, the MMR approach is implemented by the *PG-MMR* model [86] to identify salient source sentences from multi-document inputs, albeit with a different method for calculating MMR scores

from Hi-MAP; instead, ROUGE-L Recall and ROUGE-L Precision [95] serve as evaluation metrics to calculate the importance and redundancy scores. To overcome the scarcity of MDS datasets, the PG-MMR model leverages a support vector regression model that is pre-trained on an SDS dataset to recognize the important contents. This support vector regression model also calculates the score of each input sentence by considering four factors: sentence length, sentence relative/absolute position, sentence-document similarities, and sentence quality obtained by a PG network. Sentences with the top-K scores are fed into another PG network to generate a concise summary.

3.6 Transformer-based Models

As discussed, CNN-based models are not as good at processing sequential data as RNN-based models. However, RNN-based models are not amenable to parallel computing, as the current states in RNN models highly depend on results from the previous steps. Additionally, RNNs struggle to process long sequences since former knowledge will fade away during the learning process. Adopting *Transformer*-based architectures [155] is one solution to solve these problems. The Transformer is based on the self-attention mechanism, has natural advantages for parallelization, and retains relative long-range dependencies. The Transformer model has achieved promising results in MDS tasks [74, 94, 97, 98] and can replace the *DNN-based Model* in Figure 5. Most of the Transformer-based models follow an encoder-decoder structure. Transformer-based models can be divided into flat Transformer, hierarchical Transformer, and pre-trained language models.

Flat Transformer. Liu et al. [97] introduced Transformer to MDS tasks, aiming to generate a Wikipedia article from a given topic and set of references. The authors argue that the encoder-decoder-based sequence transduction model cannot cope well with long input documents, so their model selects a series of top-K tokens and feeds them into a Transformer-based decoder-only sequence transduction model to generate Wikipedia articles. More specifically, the Transformer decoder-only architecture combines the results from the extractive stage and golden summary into a sentence for training. To obtain rich semantic representation from different granularity, Jin et al. [74] proposed a Transformer-based multi-granularity interaction network *MGSum* and unified extractive and abstractive MDS. Words, sentences, and documents are considered as three granular levels of semantic unit connected by a granularity hierarchical relation graph. In the same granularity, a self-attention mechanism is used to capture the semantic relationships. Sentence granularity representation is employed in the extractive summarization, and word granularity representation is adapted to generate an abstractive summary. *MGSum* employs a fusion gate to integrate and update the semantic representation. Additionally, a spare attention mechanism is used to ensure the summary generator focus on important information. Brazinskas et al. [17] created a precedent for few-shot learning for MDS that leverages a Transformer conditional language model and a plug-in network for both extractive and abstractive MDS to overcome rapid overfitting and poor generation problems resulting from naive fine-tuning of large parameter models.

Hierarchical Transformer. To handle huge amounts of input documents (currently many large-scale MDS datasets contain more than 10,000 input document sets), Liu and Lapata [98] proposed a two-stage ***Hierarchical Transformer (HT)*** *model* with an inter-paragraph and graph-informed attention mechanism that allows the model to encode multiple input documents hierarchically instead of by simple flat concatenation. A logistic regression model is employed to select the top-K paragraphs, which are fed into a local Transformer layer to obtain contextual features. A global Transformer layer mixes the contextual information to model the dependencies of the selected paragraphs. To leverage graph structure to capture cross-document relations, Li et al. [94] proposed an end-to-end Transformer-based model *GraphSum*, based on the HT model. In the graph encoding layers, GraphSum extends the self-attention mechanism to the graph-informed self-attention

mechanism, which incorporates the graph representation into the Transformer encoding process. Furthermore, the Gaussian function is applied to the graph representation matrix to control the intensity of the graph structure's impact on the summarization model. The HT and GraphSum models are both based on the self-attention mechanism leading quadratic memory growth increases with the number of input sequences; to address this issue, Pasunuru et al. [126] modified the full self-attention with local and global attention mechanism [14] to scale the memory linearly. Dual encoders are proposed for encoding truncated concatenated documents and linearized graph information from full documents.

Pre-trained Language Models (LMs). Pre-trained Transformers on large text corpora have shown great successes in downstream NLP tasks including text summarization. The pre-trained LMs can be trained on non-summarization or SDS datasets to overcome lack of MDS data [94, 126, 178]. Most pre-trained LMs such as BERT [35] and RoBERTa [99] can work well on short sequences. In a hierarchical Transformer architecture, replacing the low-level Transformer (token-level) encoding layer with pre-trained LMs helps the model break through length limitations to perceive further information [94]. Inside a hierarchical Transformer architecture, the output vector of the "[CLS]" token can be used as input for high-level Transformer models. To avoid the self-attention quadratic-memory increment when dealing with document-scale sequences, a Longformer-based approach [14], including local and global attention mechanisms, can be incorporated with pre-trained LMs to scale the memory linearly for MDS [126]. Another solution for computational issues can be borrowed from SDS, which is to use a multi-layer Transformer architecture to scale the length of documents, allowing pre-trained LMs to encode a small block of text, and the information can be shared among the blocks between two successive layers [55]. BART [88], GPT-2 [133], and T5 [134] are pre-trained language models that can be used for language generation and they have been applied for MDS tasks [2, 122, 147]. Instead of regular language models, PEGASUS [178] is a pre-trained Transformer-based encoder-decoder model with **gap-sentences generation (GSG)** that focused on abstractive summarization. GSG shows that masking whole sentences based on importance, instead of through random or lead selection, works well for downstream summarization tasks. BART, T5, and PEGASUS are based on data-rich SDS settings. Goodwin et al. [54] evaluated these three pre-trained models on four MDS datasets and suggested that while large improvements have been made on the standard SDS task, highly abstractive MDS remains a challenge. PRIMER [164] is a pre-trained model specifically designed for MDS that can serve as a zero-shot summarizer.

3.7 Deep Hybrid Models

Many neural models can be integrated to formalize a more powerful and expressive model. In this section, we summarize the existing deep hybrid models that have proven to be effective for MDS.

CNN + LSTM + Capsule Networks. Cho et al. [29] proposed a hybrid model based on the determinantal point processes for semantically measuring sentence similarities. A convolutional layer slides over the pairwise sentences with filters of different sizes to extract low-level features. Capsule networks [138, 168] are employed to identify redundant information by transforming the spatial and orientational relationships for high-level representation. The authors also used LSTM to reconstruct pairwise sentences and add reconstruction loss to the final objective function.

CNN + Bi-LSTM + Multi-layer Perceptron (MLP). Singh et al. [145] proposed an extractive MDS framework that considers document-dependent and document-independent information. In this model, a CNN with different filters captures phrase-level representation. Full binary trees formed with these salient representations are fed to the recommended Bi-LSTM tree indexer to enable better generalization abilities. An MLP with ReLU function is employed for leaf node

transformation. More specifically, the Bi-LSTM tree indexer leverages the time serial power of LSTMs and the compositionality of recursive models to capture both semantic and compositional features.

PG Networks + Transformer. In generating a summary, it is necessary to consider the information fusion of multiple sentences, especially sentence pairs. Lebano et al. [85] found the majority of summary sentences are generated by fusing one or two source sentences, so they proposed a two-stage summarization method that considers the semantic compatibility of sentence pairs. This method joint-scores single sentences and sentence pairs to filter representative from the original documents. Sentences or sentence pairs with high scores are then compressed and rewritten to generate a summary that leverages the PG network. This article uses a Transformer-based model to encode both single sentences and sentence pairs indiscriminately to obtain the deep contextual representation of words and sequences.

3.8 The Variants of Multi-document Summarization

In this section, we briefly introduce several MDS task variants to give researchers a comprehensive understanding of MDS. These tasks can be modeled as MDS problems and adopt the aforementioned deep learning techniques and neural network architectures.

Query-oriented MDS calls for a summary from a set of documents that answers a query. It tries to solve realistic query-oriented scenario problems and only summarizes important information that best answers the query in a logical order [125]. Specifically, query-oriented MDS combines the information retrieval and MDS techniques. The content that needs to be summarized is based on the given queries. Liu and Lapata [98] incorporated the query by simply prepending the query to the top-ranked document during encoding. Pasunuru [125] involved a query encoder and integrated query embedding into an MDS model, ranking the importance of documents for a given query.

Dialogue summarization aims to provide a succinct synopsis from multiple textual utterances of two or more participants, which could help quickly capture relevant information without having to listen to long and convoluted dialogues [96]. Dialogue summary covers several areas, including meetings [45, 80, 192], email threads [180], medical dialogues [40, 75, 146], customer service [96], and media interviews [191]. Challenges in dialogue summarization can be summarized into the following seven categories: informal language use, multiple participants, multiple turns, referral and coreference, repetition and interruption, negations and rhetorical questions, and role and language change [25]. The flow of the dialogue would be neglected if MDS models are directly applied for dialogue summarization. Liu et al. [96] relied on human annotations to capture the logic of the dialogue. Wu et al. [163] used summary sketch to identify the interaction between speakers and their corresponding textual utterances in each turn. Chen and Yang [25] proposed a multi-view sequence-to-sequence-based encoder to extract dialogue structure and a multi-view decoder to incorporate different views to generate final summaries.

Stream summarization aims to summarize new documents in a continuously growing document stream, such as information from social media. Temporal summarization and **real-time summarization (RTS)**[9] can be seen as a form of stream document summarization. Stream summarization considers both historical dependencies and future uncertainty of the document stream. Yang et al. [167] used deep reinforcement learning to solve the relevance, redundancy, and timeliness issues in steam summarization. Tan et al. [150] transformed the real-time

[9] http://trecrts.github.io/.

summarization task as a sequential decision-making problem and used an LSTM layer and three fully connected neural network layers to maximize the long-term rewards.

3.9 Discussion

In this section, we have reviewed the state-of-the-art works of deep-learning-based MDS models according to the neural networks applied. Table 2 summarizes the reviewed works by considering the type of neural networks, construction types, and concatenation methods and provides a high-level summary of their relative advantages and disadvantages. Transformer-based models have been most commonly used in the last 3 years because they overcome the limitations of CNN's fixed-size receptive field and RNN's inability to parallel process. However, deep-learning-based MDS models face some challenges. First, the complexity of deep-learning-based models and the data-driven deep learning systems do require more training data, with concomitant increased efforts in data labeling, and more computing resources than non-deep-learning-based methods. Inevitably, deep-learning-based MDS models require more computation during the training phase. During the inference process, they generally consume more computation power than non-deep-learning-based methods as well. Second, deep-learning-based methods lack linguistic knowledge that can serve as important roles in assisting deep-learning-based learners to have informative representation and better guide the summary generation. We believe that this is one possible reason that some non-deep-learning-based MDS methods sometimes show better performance than deep-learning-based methods [21, 101] as non-deep-learning-based methods pay more attention to linguistic information. We discuss this point in Section 7.1. Further research could also be based on techniques adopted in non-deep-learning-based MDS as reviewed in [39, 46, 142]. Third, deep-learning-based models can be regarded as black boxes with high non-linearity. It is challenging to understand the detailed transformation inside them. Exploring the interpretability of MDS models allows researchers to understand the effect of each module in MDS neural models, therefore guiding the model design with a more accurate target. However, there is little existing work on the interpretability of MDS models, which is of great help in improving the quality of summaries. More discussions about explainable deep learning MDS models can be found in Section 7.6.

4 OBJECTIVE FUNCTIONS

In this section, we will take a closer look at different objective functions adopted by various MDS models. In summarization models, objective functions play an important role by guiding the model to achieve specific purposes. To the best of our knowledge, we are the first to provide a comprehensive survey on different objectives of summarization tasks.

4.1 Cross-entropy Objective

Cross-entropy usually acts as an objective function to measure the distance between two distributions. Many existing MDS models adopt it to measure the difference between the distributions of generated summaries and the golden summaries [20, 29, 160, 172, 177, 184]. Formally, the cross-entropy loss is defined as

$$L_{CE} = -\sum_{i=1} \mathbf{y_i} \log(\hat{\mathbf{y}}_\mathbf{i}), \tag{1}$$

where $\mathbf{y_i}$ is the target score from golden summaries and machine-generated summaries, and $\hat{\mathbf{y}}_\mathbf{i}$ is the predicted estimation from the deep-learning-based models. Different from calculations in other tasks, such as text classification, in summarization tasks, $\mathbf{y_i}$ and $\hat{\mathbf{y}}_\mathbf{i}$ have several methods to calculate. $\hat{\mathbf{y}}_\mathbf{i}$ usually is calculated by ROUGE (refer to Section 5). For example, ROUGE-1 [6], ROUGE-2 [98], or the normalized average of ROUGE-1 and ROUGE-2 scores [173] could be adopted to compute the ground-truth score between the selected sentences and golden summary.

Table 2. Deep-learning-based Methods

Methods	Works	Construction Types			Document-level Relationship		Comparison of DL-based techniques
		Ext	Abs	Hyb	FC	HC	Pros and Cons
RNN	MeanSum [31]		✓		✓		Pros: Can capture sequential relations and syntactic/semantic information from word sequences Cons: Not easy for parallel computing; highly depending on results from the previous steps
	Zhang et al. [177]		✓		✓		
	STDS [187]	✓				✓	
	ParaFuse_doc [114]		✓			✓	
	R2N2 [20]	✓			✓		
	CondaSum [4]			✓		✓	
	C-Attention [93]		✓		✓		
	Wang and Ling[162]		✓		✓		
	RL-MMR [105]	✓			✓		
	Coavoux et al. [33]		✓		✓		
CNN	MV-CNN [184]	✓			✓		Pros: Good parallel computing; Cons: Not good at processing sequential data
	TCSum [19]	✓			✓		
	CNNLM [174]	✓			✓		
	PriorSum [21]	✓			✓		
	Angelidis and Lapata [5]	✓			✓		
GNN	Yasunaga et al. [173]	✓				✓	Pros: Can capture cross-document and in-document relations Cons: Inefficient when dealing with large graphs
	SemSentSum [6]	✓				✓	
	Scisummnet [172]	✓				✓	
	HDSG [160]	✓				✓	
PG	PG-MMR [86]		✓		✓		Pros: Low redundancy Cons: Hard to train
	Hi-MAP [44]		✓		✓		
Transformer	HT [98]		✓			✓	Pros: Good performance; good parallel computing; can capture cross-document and in-document relations Cons: Time-consuming; problems with position encoding
	MGSum [74]	✓	✓			✓	
	FewSum [17]	✓	✓		✓		
	GraphSum [94]		✓			✓	
	Bart-Long [126]		✓			✓	
	WikiSum [97]			✓	✓		
Deep Hybid Model	Cho et al. [29]	✓			✓		Pros: Combines the advantages of different DL models Cons: Computationally intensive
	GT-SingPairMix [85]		✓		✓		
	HNet [145]	✓			✓		

"Ext", "Abs" and "Hyd" mean extractive, abstractive and hybrid respectively; "FC" and "HC" represent Flat Concatenate, Hierarchical Concatenate respectively.

4.2 Reconstructive Objective

Reconstructive objectives are used to train a distinctive representation learner by reconstructing the input vectors in an unsupervised learning manner. The objective function is defined as

$$L_{Rec} = \left\| \mathbf{x_i} - \phi'(\phi(\mathbf{x_i}; \theta); \theta') \right\|_*, \tag{2}$$

where $\mathbf{x_i}$ represents the input vector; ϕ and ϕ' represent the encoder and decoder with θ and θ' as their parameters, respectively; $||\cdot||_*$ represents norm (* stands for 0, 1, 2, ..., infinity); and L_{Rec} is a measuring function to calculate the distance between source documents and their reconstructive outputs. Chu and Liu [31] used a reconstructive loss to constrain the generated text into the natural language domain, reconstructing reviews in a token-by-token manner. Moreover, this paper also proposes a variant termed *reconstruction cycle loss*. By using the variant, the reviews are encoded into a latent space to further generate the summary, and the summary is then decoded to the reconstructed reviews to form another reconstructive closed loop. An unsupervised learning loss was designed by Li et al. [93] to reconstruct the condensed output vectors to the original input sentence vectors with L_2 distance. This paper further constrains the condensed output vector with a L_1 regularizer to ensure sparsity. Similarly, Zheng et al. [187] adopted a bi-directional GRU encoder-decoder framework to reconstruct both news sentences and comment sentences in a word sequence manner. Liu et al. [97] concatenated both input and output sequences to predict the next

token to train the abstractive model. There are also some variants, such as leveraging the latent vectors of variational auto-encoder for reconstruction, to capture better representation. Li et al. [92] introduced three individual reconstructive losses to consider both news reconstruction and comment reconstruction separately, along with a variational auto-encoder lower bound. Brazinskas et al. [16] utilized a variational auto-encoder to generate the latent vectors of given reviews, where each review is reconstructed by the latent vectors combined with other reviews.

4.3 Redundancy Objective

Redundancy is an important objective to minimize the overlap between semantic units in a machine-generated summary. By using this objective, models are encouraged to maximize information coverage. Formally,

$$L_{Red} = Sim(\mathbf{x_i}, \mathbf{x_j}), \tag{3}$$

where $Sim(\cdot)$ is the similarity function to measure the overlap between different $\mathbf{x_i}$ and $\mathbf{x_j}$, which can be phrases, sentences, topics, or documents. The redundancy objective is often treated as an auxiliary objective combined with other loss functions. Li et al. [93] penalized phrase pairs with similar meanings to eliminate the redundancy. Nayeem et al. [114] used the redundancy objective to avoid generating repetitive phrases, constraining a sentence to appear only once while maximizing the scores of important phrases. Zheng et al. [187] adopted a redundancy loss function to measure overlaps between subtopics; intuitively, smaller overlaps between subtopics resulted in less redundancy in the output domain. Yin and Pei [174] proposed a redundancy objective to estimate the diversity between different sentences.

4.4 Max Margin Objective

Max Margin Objectives (MMOs) are also used to empower the MDS models to learn better representation. The objective function is formalized as

$$L_{Margin} = \max\left(0, f(\mathbf{x_i}; \theta) - f(\mathbf{x_j}; \theta) + \gamma\right), \tag{4}$$

where $\mathbf{x_i}$ and $\mathbf{x_j}$ represent the input vectors, θ are parameters of the model function $f(\cdot)$, and γ is the margin threshold. The MMO aims to force function $f(\mathbf{x_i}; \theta)$ and function $f(\mathbf{x_j}; \theta)$ to be separated by a predefined margin γ. In Cao et al. [19], an MMO is designed to constrain a pair of randomly sampled sentences with different salience scores—the one with a higher score should be larger than the other one more than a marginal threshold. Two max margin losses are proposed in Zhong et al. [188]: a margin-based triplet loss that encouraged the model to pull the golden summaries semantically closer to the original documents than to the machine-generated summaries, and a pair-wise margin loss based on a greater margin between paired candidates with more disparate ROUGE score rankings.

4.5 Multi-task Objective

Supervision signals from MDS objectives may not be strong enough for representation learners, so some works seek other supervision signals from multiple tasks. A general form is as follows:

$$L_{Mul} = L_{Summ} + L_{Other}, \tag{5}$$

where L_{Summ} is the loss function of MDS tasks, and L_{Other} is the loss function of an auxiliary task. Angelidis and Lapata [5] assumed that the aspect-relevant words provide not only a reasonable basis for model aspect reconstruction but also a good indicator for product domain. Similarly, multi-task classification was introduced by Cao et al. [19]. Two models are maintained: text classification and text summarization models. In the first model, CNN is used to classify text categories and cross-entropy loss is used as the objective function. The summarization model and the text classification

model share parameters and pooling operations, so are equivalent to the shared document vector representation. Coavoux et al. [33] jointly optimized the model from a language modeling objective and two other multi-task supervised classification losses, which are polarity loss and aspect loss.

4.6 Other Types of Objectives

There are many other types of objectives in addition to those mentioned above. Cao et al. [21] proposed using ROUGE-2 to calculate the sentence saliency scores, and the model tries to estimate this saliency with linear regression. Yin and Pei [174] suggested summing the squares of the prestige vectors calculated by the PageRank algorithm to identify sentence importance. Zhang et al. [184] proposed an objective function by ensembling individual scores from multiple CNN models; besides the cross-entropy loss, a consensus objective is adopted to minimize disagreement between each pair of classifiers. Amplayo and Lapata [4] used two objectives in the abstract module: the first to optimize the generation probability distribution by maximizing the likelihood, and the second to constrain the model output to be close to its golden summary in the encoding space, as well as being distant from the random sampled negative summaries. Chu and Liu [31] designed a similarity objective that shares the encoder and decoder weights within the auto-encoder module, while in the summarization module, the average cosine distance indicates the similarity between the generated summary and the reviews. A variant similarity objective termed *early cosine objective* is further proposed to compute the similarity in a latent space that is the average of the cell states and hidden states to constrain the generated summaries semantically close to reviews.

4.7 Discussion

In MDS, cross-entropy is the most commonly adopted objective function that bridges the predicted candidate summaries and the golden summaries by treating the golden summaries as strong supervision signals. However, adopting cross-entropy loss alone may not lead the model to achieve good performance since the supervisory signal for cross-entropy objective is not strong enough by itself to effectively learn good representation. Several other objectives can thus serve as complements; e.g., reconstruction objectives offer a view from the unsupervised learning perspective; the redundancy objective constrains models from generating redundant content; and max-margin objectives require step-change improvements from previous versions. By using multiple objectives, model optimization could be conducted with the input documents themselves if the manual annotation is scarce. The models that adopt multi-task objectives explicitly define multiple auxiliary tasks to assist the main summarization task for better generalization and provide various constraints from different angles that lead to better model optimization.

5 EVALUATION METRICS

Evaluation metrics are used to measure the effectiveness of a given method objectively, so well-defined evaluation metrics are crucial to MDS research. We classify the existing evaluation metrics into two categories and will discuss each category in detail: (1) ROUGE: the most commonly used evaluation metrics in the summarization community, and (2) other evaluation metrics that have not been widely used in MDS research to date.

5.1 ROUGE

ROUGE [95] is a collection of evaluation indicators that is one of the most essential metrics for many natural language processing tasks, including machine translation and text summarization. ROUGE obtains prediction/ground-truth similarity scores through comparing automatically generated summaries with a set of corresponding human-written references. ROUGE has many

variants to measure candidate abstracts in a variety of ways [95]. The most commonly used ones are ROUGE-N and ROUGE-L.

ROUGE-N (*ROUGE with n-gram co-occurrence statistics*) measures an n-gram recall between a reference and their corresponding candidate summaries [95]. ROUGE-N can be calculated as

$$ROUGE\text{-}N = \frac{\sum_{Sum \in \{Ref\}} \sum_{gram_n \in Sum} Count_{match}(gram_n)}{\sum_{Sum \in \{Ref\}} \sum_{gram_n \in Sum} Count(gram_n)}, \tag{6}$$

where Ref and Sum are reference and machine-generated summary, n represents the length of n-gram, and $Count_{match}(gram_n)$ represents the maximum number of n-grams in the reference summary and corresponding candidates. The numerator of ROUGE-N is the number of n-grams owned by both the reference and generated summary, while the denominator is the total number of n-grams occurring in the golden summary. The denominator could also be set to the number of candidate summary n-grams to measure precision; however, ROUGE-N mainly focuses on quantifying recall, so precision is not usually calculated. ROUGE-1 and ROUGE-2 are special cases of ROUGE-N that are usually chosen as best practices and represent the unigram and bigram.

ROUGE-L (*ROUGE with Longest Common Subsequence*) adopts the longest common subsequence algorithm to count the longest matching vocabularies [95]. Formally, ROUGE-L is calculated using

$$F_{lcs} = \frac{(1+\beta^2) R_{lcs} P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}}, \tag{7}$$

where

$$R_{lcs} = \frac{LCS(Ref, Sum)}{m}, \tag{8}$$

and

$$P_{lcs} = \frac{LCS(Ref, Sum)}{n}, \tag{9}$$

where LCS(·) represents the longest common subsequence function. ROUGE-L is termed as an LCS-based F-measure as it is obtained from LCS-Precision P_{lcs} and LCS-Recall R_{lcs}. β is the balance factor between R_{lcs} and P_{lcs}. It can be set by the fraction of P_{lcs} and R_{lcs}; by setting β to a big number, only R_{lcs} is considered. The use of ROUGE-L enables the measurement of the similarity of two text sequences at sentence level. ROUGE-L also has the advantage of deciding the n-gram without extra manual input, since the calculation of LCS can count grams adaptively.

Other ROUGE-based Metrics. *ROUGE-W* [95] is proposed to weight consecutive matches to better measure semantic similarities between two texts. *ROUGE-S* [95] stands for ROUGE with Skip-bigram co-occurrence statistics that allows the bigram to skip arbitrary words. An extension of ROUGE-S, *ROUGE-SU* [95] refers to ROUGE with Skip-bigram plus Unigram-based co-occurrence statistics and is able to be obtained from ROUGE-S by adding a beginning-of-sentence token at the start of both references and candidates. *ROUGE-WE* [119] is proposed to further extend ROUGE by measuring the pair-wise summary distances in word embedding space. In recent years, more ROUGE-based evaluation models have been proposed to compare golden and machine-generated summaries, not just according to their literal similarity, but also considering semantic similarity [141, 181, 186]. In terms of the ROUGE metric for multiple golden summaries, the Jackknifing procedure (similar to K-fold validation) has been introduced [95]. The M best scores are computed from sets composed of M-1 reference summaries, and the final ROUGE-N is the average of M scores. This procedure can also be applied to ROUGE-L, ROUGE-W, and ROUGE-S.

5.2 Other Evaluation Metrics

Besides *ROUGE*-based [95] metrics, other evaluation metrics for MDS exist but have received less attention than ROUGE. We hope this section will give researchers and practitioners a holistic

view of alternative evaluation metrics in this field. Based on the mode of summary matching, we divide the evaluation metrics into two groups: lexical matching metrics and semantic matching metrics.

Lexical Matching Metrics. *BLEU* [123] is a commonly used vocabulary-based evaluation metric that provides a precision-based evaluation indicator, as opposed to ROUGE that mainly focuses on recall. *Perplexity* [72] is used to evaluate the quality of the language model by calculating the negative log probability of a word's appearance. A low perplexity on a test dataset is a strong indicator of a summary's high grammatical quality because it measures the probability of words appearing in sequences. Based on *Pyramid* [117] calculation, the abstract sentences are manually divided into several **Summarization Content Units (SCUs)**, each representing a core concept formed from a single word or phrase/sentence. After sorting SCUs in order of importance to form the *Pyramid*, the quality of automatic summarization is evaluated by calculating the number and importance of SCUs included in the document [118]. Intuitively, more important SCUs exist at higher levels of the pyramid. Although *Pyramid* shows a strong correlation with human judgment, it requires professional annotations to match and evaluate SCUs in generated and golden summaries. Some recent works focus on the construction of *Pyramid* [48, 62, 124, 143, 169]. *Responsiveness* [100] measures content selection and linguistic quality of summaries by directly rating scores. Additionally, the assessments are calculated without reference to model summaries. *Data Statistics* [56] contain three evaluation metrics: extractive fragment coverage measures the novelty of generated summaries by calculating the percentage of words in the summary that are also present in source documents; extractive fragment density measures the average length of the extractive block to which each word in the summary belongs; and compression ratio compares the word numbers in the source documents and generated summary.

Semantic Matching Metrics. Metric for Evaluation of Translation with Explicit Ordering (*METEOR*) [10] is an improvement to BLEU. The main idea behind METEOR is that while candidate summaries can be correct with similar meanings, they are not exactly matched with references. In such a case, WordNet[10] is introduced to expand the synonym set, and the word form is also taken into account. *SUPERT* [49] is an unsupervised MDS evaluation metric that measures the semantic similarity between the pseudo-reference summary and the machine-generated summary. *SUPERT* obviates the need for human annotations by not referring to golden summaries. Contextualized embeddings and soft token alignment techniques are leveraged to select salient information from the input documents to evaluate summary quality. *Preferences-based Metric* [194] is a pairwise sentence preference-based evaluation model and it does not depend on the golden summaries. The underlying premise is to ask annotators about their pair-wise preferences rather than writing complex golden summaries, and they are much easier and faster to obtain than traditional reference summary-based evaluation models. *BERTScore* [181] computes a similarity score for each token within the candidate sentence and the reference sentence. It measures the soft overlap of two texts' BERT embeddings. *MoverScore* [186] adopts a distance to evaluate the agreement between two texts in the context of BERT and ELMo word embeddings. This proposed metric has a high correlation with human judgment of text quality by adopting the earth mover's distance. *Importance* [129] is a simple but rigorous evaluation metric from the aspect of information theory. It is a final indicator calculated from the three aspects: *Redundancy*, *Relevance*, and *Informativeness*. A good summary should have low *Redundancy* and high *Relevance* and high *Informativeness*. The cluster of *Human Evaluation* is used to supplement automatic evaluation on relatively small instances. Annotators evaluate the quality of machine-generated summaries by rating

[10] https://wordnet.princeton.edu/.

Informativeness, Fluency, Conciseness, Readability, Relevance. Model ratings are usually computed by averaging the rating on all selected summary pairs.

5.3 Discussion

We summarize the advantages and disadvantages of above-mentioned evaluation metrics in Table 3. Although there are many evaluation metrics for MDS, the indicators of the ROUGE series are generally accepted by the summarization community. Almost all research works utilize ROUGE for evaluation, while other evaluation indicators are just for assistance currently. Among the ROUGE family, ROUGE-1, ROUGE-2, and ROUGE-L are the most commonly used evaluation metrics. In addition, there are plenty of existing evaluation metrics in other natural language processing tasks that could be potentially adjusted for MDS tasks, such as *efficiency*, *effectiveness,* and *coverage* from information retrieval.

6 DATASETS

Compared to SDS tasks, large-scale MDS datasets, which contain more general scenarios with many downstream tasks, are relatively scarce. In this section, we present our investigation on the 10 most representative datasets commonly used for MDS and its variant tasks.

DUC and TAC. Document Understanding Conference (DUC)[11] provides official text summarization competitions each year from 2001 to 2007 to promote summarization research. DUC changed its name to **Text Analysis Conference (TAC)**[12] in 2008. Here, the DUC datasets refer to the data collected from 2001 to 2007; the TAC datasets refer to the datasets after 2008. Both DUC and TAC are from the news domains, including various topics such as politics, natural disasters, and biography. Nevertheless, as shown in Table 4, the DUC and TAC datasets provide small datasets for model evaluation that only include hundreds of news documents and human-annotated summaries. Of note, the first sentence in a news item is usually information-rich and renders bias in the news datasets, so it fails to reflect the structure of natural documents in daily lives. These two datasets are on a relatively small scale and not ideal for large-scale deep-neural-based MDS model training and evaluation.

OPOSUM. OPOSUM [5] collects multiple reviews of six product domains from Amazon. This dataset contains not only multiple reviews and corresponding summaries but also products' domain and polarity information. The latter information could be used as auxiliary supervision signals.

WikiSum. WikiSum [97] targets abstractive MDS. For a specific Wikipedia theme, the documents cited in Wikipedia articles or the top-10 Google search results (using the Wikipedia theme as a query) are seen as the source documents. Golden summaries are the real Wikipedia articles. However, some of the URLs are not available and can be identical to each other in parts. To remedy these problems, Liu and Lapata [98] cleaned the dataset and deleted duplicated examples, so here we report statistical results from [98].

Multi-News. Multi-News [44] is a relatively large-scale dataset in the news domain; the articles and human-written summaries are all from the web.[13] This dataset includes 56,216 article-summary pairs and contains trace-back links to the original documents. Moreover, the authors compared the Multi-News dataset with prior datasets in terms of coverage, density, and compression, revealing that this dataset has various arrangement styles of sequences.

[11] http://duc.nist.gov/.
[12] http://www.nist.gov/tac/.
[13] http://newser.com.

Table 3. Advantages and Disadvantages of Different Evaluation Metrics

Evaluation Metrics		Advantages	Disadvantages
Lexical Matching Metrics	ROUGE	• Widely used • Intuitive • Easily computed	• Cannot measure texts semantically • Exact matching
	BLEU	• Intuitive • Easily computed • High correlations with human judgments	• Cannot measure texts semantically • Cannot deal with languages lacking word boundaries
	Perplexity	• Easily computed • Intuitive	• Sensitive to certain symbols and words
	Pyramid	• High correlations with human judgments	• Requires manual extraction of units • Bias results easily
	Responsiveness	• Consider both content and linguistic quality • Can be calculated without reference	• Not widely adopted
	Data Statistics	• Can measure the density and coverage of summary	• Cannot measure texts semantically
Semantic Matching Metrics	METEOR	• Consider non-exact matching	• Sensitive to length
	SUPERT	• Can measure texts semantic similarity	• Not widely adopted
	Preferences-based Metric	• Does not depend on the golden summaries	• Require human annotations
	BERTScore	• Semantically measure texts to some extent • Mimic human evaluation	• High computational demands
	MoverScore	• Semantically measure texts to some extent • More similar to human evaluation by adopting earth mover's distance	• High computational demands
	Importance	• Combining redundancy, relevance, and informativeness • Theoretically supported	• Non-trivial for implementation
	Human Evaluation	• Can accurately and semantically measure texts	• Require human annotations

Opinosis. The Opinosis dataset [47] contains reviews of 51 topic clusters collected from TripAdvisor,[14] Amazon,[15] and Edmunds.[16] For each topic, approximately 100 sentences on average are provided and the reviews are fetched from different sources. For each cluster, five professionally written golden summaries are provided for model training and evaluation.

[14] https://www.tripadvisor.com/.
[15] https://www.amazon.com.au/.
[16] https://www.edmunds.com/.

Table 4. Comparison of Different Datasets

Datasets	Cluster #	Document #	Summ #	Ave Summ Len	Topic
DUC01	30	309 docs	60 summ	100 words	News
DUC02	59	567 docs	116 summ	100 words	News
DUC03	30	298 docs	120 summ	100 words	News
DUC04	50	10 docs/cluster	200 summ	665 bytes	News
DUC05	50	25–50 docs/cluster	140 summ	250 words	News
DUC06	50	25 docs/cluster	4 summ/cluster	250 words	News
DUC07	45	25 docs/cluster	4 summ/cluster	250 words	News
TAC 2008	48	10 docs/cluster	4 summ/cluster	100 words	News
TAC 2009	44	10 docs/cluster	4 summ/cluster	100 words	News
TAC 2010	46	10 docs/cluster	4 summ/cluster	100 words	News
TAC 2011	44	10 docs/cluster	4 summ/cluster	100 words	News
OPOSUM	60	600 rev	1 summ/cluster	100 words	Amazon reviews
WikiSum	-	train/val/test 1,579,360/38,144/38,205	1 summ/cluster	139.4 tokens/ summ	Wikipedia
Multi-News	-	train/val/test 44,972/5,622/5,622 2–10 docs/cluster	1 summ/cluster	263.66 words/summ 9.97 sents/summ 262 tokens/summ	News
Opinosis	51	6,457 rev	5 summ/cluster	-	Site reviews
Rotten Tomatoes	3731	99.8 rev/cluster	1 summ/cluster	19.6 tokens/summ	Movie reviews
Yelp	-	train/val/test bus: 10,695/1,337/1,337 rev: 1,038,184/129,856/129,840	-	-	Customer reviews
Scisumm	1000	21–928 cites/paper 15 sents/refer	1 summ/cluster	151 words	Science Paper
WCEP	10200	235 docs/cluster	1 summ/cluster	32 words	Wikipedia
Multi-XScience	-	train/val/test 30,369/5,066/5,093	1 summ/cluster	116.44 words/summ	Science Paper

In the table, "Ave", "Summ", "Len", "bus","rev" and "#" represent average, summary, length, business, reviews and numbers respectively; "Docs" and "sents" mean documents and sentences respectively.

Rotten Tomatoes. The Rotten Tomatoes dataset [162] consists of the collected reviews of 3,731 movies from the Rotten Tomato website.[17] The reviews contain both professional critics and user comments. For each movie, a one-sentence summary is created by professional editors.

Yelp. Chu and Liu [31] proposed a dataset named Yelp based on the Yelp Dataset Challenge. This dataset includes multiple customer reviews with five-star ratings. The authors provided 100 manually written summaries for model evaluation using **Amazon Mechanical Turk (AMT)**, within which every eight input reviews are summarized into one golden summary.

Scisumm. The Scisumm dataset [172] is a large, manually annotated corpus for scientific document summarization. The input documents are a scientific publication, called the reference paper, and multiple sentences from the literature that cite this reference paper. In the SciSumm dataset, the 1,000 most cited papers from the ACL Anthology Network [132] are treated as reference papers, and an average of 15 citation sentences are provided after cleaning. For each cluster, one golden summary is created by five NLP-based PhD students or equivalent professionals.

WCEP. The **Wikipedia Current Events Portal (WCEP)** dataset [51] contains human-written summaries of recent news events. Similar articles are provided by searching similar articles from

[17] http://rottentomatoes.com.

the Common Crawl News dataset[18] to extend the inputs to obtain large-scale news articles. Overall, the WCEP dataset has good alignment with real-world industrial use cases.

Multi-XScience. The source data of Multi-XScience [101] are from Arxiv and Microsoft academic graphs, and this dataset is suitable for abstractive MDS. Multi-XScience contains fewer positional and extractive biases than the WikiSum and Multi-News datasets, so the drawback of obtaining higher scores from a copy sentence at a certain position can be partially avoided.

Datasets for MDS Variants. The representative query-oriented MDS datasets are Debatepedia [115], AQUAMUSE [82], and QBSUM [185]. The representative dialogue summarization datasets are DIALOGSUM [26], AMI [24], MEDIASUM [191], and QMSum [189]. RTS is a track at the **Text Retrieval Conference (TREC)** that provides several RTS datasets.[19] The Tweet Contextualization track [13] (2012–2014) is derived from the INEX 2011 Question Answering Track and focuses on more NLP-oriented tasks and moves to MDS.

Discussion. Table 4 compares 20 MDS datasets based on the numbers of clusters and documents, the number and the average length of summaries, and the field to which the dataset belongs. Currently, the main areas covered by the MDS datasets are news (60%), scientific papers (10%), and Wikipedia (10%). In the early development of the MDS, most studies were performed on the DUC and TAC datasets. However, the size of these datasets is relatively small, and thus not highly suitable for training deep neural network models. Datasets on news articles are also common, but the structure of news articles (highly compressed information in the first paragraph or first sentence of each paragraph) can cause positional and extractive biases during training. In recent years, large-scale datasets such as WikiSum and Multi-News have been developed and used by researchers to meet training requirements, reflecting the rising trend of data-driven approaches.

7 FUTURE RESEARCH DIRECTIONS AND OPEN ISSUES

Although existing works have established a solid foundation for MDS, it is a relatively understudied field compared with SDS and other NLP topics. Summarizing on multi-modal data, medical records, codes, project activities, and MDS combining with Internet of Things [183] have still received less attention. Actually, MDS techniques are beneficial for a variety of practical applications, including generating Wikipedia articles and summarizing news, scientific papers, and product reviews, and individuals and industries have a huge demand for compressing multiple related documents into high-quality summaries. This section outlines several prospective research directions and open issues that we believe are critical to resolving in order to advance the field.

7.1 Capturing Cross-document Relations for MDS

Currently, many MDS models still center on a simple concatenation of input documents into a flat sequence, ignoring cross-document relations. Unlike SDS, MDS input documents may contain redundant, complementary, or contradictory information [130]. Discovering cross-document relations, which can assist models to extract salient information, improves the coherence and reduces redundancy of summaries [94]. Research on capturing cross-document relations has begun to gain momentum in the past 2 years; one of the most widely studied topics is *graphical models*, which can easily be combined with deep-learning-based models such as graph neural networks and Transformer models. Several existing works indicate the efficacy of graph-based deep learning models in capturing semantic-rich and syntactic-rich representation and generating high-quality summaries [94, 160, 172, 173]. To this end, a promising and important direction would be to design a better

[18] https://commoncrawl.org/2016/10/news-dataset-available/.
[19] http://trecrts.github.io/.

mechanism to introduce different graph structures [30] or linguistic knowledge [15, 103], possibly into the attention mechanism in deep-learning-based models, to capture cross-document relations and to facilitate summarization.

7.2 Creating More High-quality Datasets for MDS

Benchmark datasets allow researchers to train, evaluate, and compare the capabilities of different models at the same stage. High-quality datasets are critical to developing MDS tasks. DUC and TAC, the most common datasets used for MDS tasks, have a relatively small number of samples so are not very suitable for training DNN models. In recent years, some large datasets have been proposed, including WikiSum [97], Multi-News [44], and WCEP [51], but more efforts are still needed. Datasets with documents of rich diversity, with minimal positional and extractive biases, are desperately required to promote and accelerate MDS research, as are datasets for other applications such as summarization of medical records or dialogue [111], email [154, 176], code [106, 135], software project activities [3], legal documents [76], and multi-modal data [89]. The development of large-scale cross-task datasets will facilitate multi-task learning [165]. However, the datasets of MDS combining with text classification, question answering, or other language tasks have seldom been proposed in the MDS research community, but these datasets are essential and widely employed in industrial applications.

7.3 Improving Evaluation Metrics for MDS

To our best knowledge, there are no evaluation metrics specifically designed for MDS models—SDS and MDS models share the same evaluation metrics. New MDS evaluation metrics should be able to (1) evaluate the relations between the different input documents in the generated summary, (2) measure to what extent the redundancy in input documents is reduced, and (3) judge whether the contradictory information across documents is reasonably handled. A good evaluation indicator is able to reflect the true performance of an MDS model and guide design of improved models. However, current evaluation metrics [43] still have several obvious defects. For example, despite the effectiveness of commonly used ROUGE metrics, they struggle to accurately measure the semantic similarity between a golden and generated summary because ROUGE-based evaluation metrics only consider vocabulary-level distances; as such, even if a ROUGE score improves, it does not necessarily mean that the summary is of a higher quality and so is not ideal for model training. Recently, some works extend ROUGE along with WordNet [141] or pre-trained LMs [181] to alleviate these drawbacks. It is challenging to propose evaluation indicators that can reflect the true quality of generated summaries comprehensively and as semantically as human raters. Another frontline challenge for evaluation metrics research is unsupervised evaluation, which is was explored by a number of recent studies [49, 148].

7.4 Reinforcement Learning for MDS

Reinforcement learning [110] is a cluster of algorithms based on dynamic programming according to the Bellman Equation to deal with sequential decision problems, where state transition dynamics of the environment are provided in advance. Several existing works [113, 127, 171] model the document summarization task as a sequential decision problem and adopt reinforcement learning to tackle the task. Although deep reinforcement learning for SDS has made great progress, we still face challenges to adapt existing SDS models to MDS, as the latter suffers from a large state, action space, and problems with high redundancy and contradiction [105]. Additionally, current summarization methods are based on model-free reinforcement learning algorithms, in which the model is not aware of environment dynamics but continuously explores the environment through simple trial-and-error strategies, so they inevitably suffer from low sampling efficiencies. Nevertheless,

the model-based approaches can leverage data more efficiently since they update models upon the prior to the environment. In this case, data-efficient reinforcement learning for MDS could potentially be explored in the future.

7.5 Pre-trained Language Models for MDS

In many NLP tasks, the limited labeled corpora are not adequate to train semantic-rich word vectors. Using large-scale, unlabeled, task-agnostic corpora for pre-training can enhance the generalization ability of models and accelerate convergence of networks [109, 128]. At present, pre-trained LMs have led to successes in many deep-learning-based NLP tasks. Among the reviewed papers [2, 85, 94, 122, 147, 188], multiple works adopt pre-trained LMs for MDS and achieve promising improvements. Applying pre-trained LMs such as BERT [35], GPT-2 [133], GPT-3 [18], XLNet [170], ALBERT [84], or T5 [134] and fine-tuning them on a variety of downstream tasks allow the model to achieve faster convergence speed and can improve model performance. MDS requires the model to have a strong ability to process long sequences. It is promising to explore powerful LMs specifically targeting long sequence input characteristics and avoiding quadratic memory growth for self-attention mechanisms, such as Longformer [14], REFORMER [79], or Big Bird [175] with pre-trained models. Also, tailor-designed pre-trained LMs for summarization have not been well explored; e.g., using gap sentences generation is more suitable than using masked language models [178]. Most MDS methods focus on combining pre-trained LMs in an encoder, and as for capturing cross-document relations, applying them in a decoder is also a worthwhile direction for research [126]. Other promising directions in this area involve exploring pre-trained LMs in languages other than English and specialized LMs for dealing with specific summarization tasks, e.g., LMs pre-trained on scientific articles.

7.6 Creating Explainable Deep Learning Model for MDS

Researchers are more focused on designing deep architectures toward a certain MDS task by improving the models' performance while ignoring their interpretabilities. However, an explainable model can reveal how it generates candidate summaries—to distinguish whether the model has learned the distribution of generating condensed and coherent summaries from multiple documents without bias—and is thus crucial for model building. Recently, a large number of research works into explainable models [136, 179] have proposed easing the non-interpretable concern of deep neural networks, within which model attention plays an especially important role in model interpretation [140, 190]. While explainable methods have been intensively researched in NLP [68, 83], studies into explainable MDS models are relatively scarce and would benefit from future development.

7.7 Adversarial Attack and Defense for MDS

Adversarial examples are strategically modified samples that aim to fool deep-neural-network-based models. An adversarial example is created via the worst-case perturbation of the input to which a robust DNN model would still assign correct labels, while a vulnerable DNN model would have high confidence in the wrong prediction. The idea of using adversarial examples to examine the robustness of a DNN model originated from research in Computer Vision [149] and was introduced in NLP by Jia and Liang [73]. An essential purpose for generating adversarial examples for neural networks is to utilize these adversarial examples to enhance the model's robustness [53]. Therefore, research on adversarial examples not only helps identify and apply a robust model but also helps to build robust models for different tasks. Following the pioneering work proposed by Jia and Liang [73], many attack methods have been proposed to address this problem in NLP

applications [182] with limited research for MDS [28]. It is worth filling this gap by exploring existing and developing new adversarial attacks on the state-of-the-art DNN-based MDS models.

7.8 Multi-modality for MDS

Existing multi-modal summarization is based on non-deep-learning techniques [69–71, 90], leaving a huge opportunity to exploit deep learning techniques for this task. Multi-modal learning has led to successes in many deep learning tasks, such as Visual Language Navigation [161] and Visual Question Answering [7]. Combining MDS with multi-modality has a range of applications:

- Text + image: generating summaries with pictures and texts for documents with pictures. This kind of multi-modal summary can improve the satisfaction of users [193].
- Text + video: based on the video and its subtitles, generating a concise text summary that describes the main context of video [121]. Movie synopsis is one application.
- Text + audio: generating short summaries of audio files that people could quickly preview without actually listening to the entire audio recording [42].

Deep learning is well suited for multi-modal tasks [58], as it is able to effectively capture highly nonlinear relationships between images, text, or video data. Existing MDS models target dealing with textual data only. Involving richer modalities based on textual data requires models to embrace larger capacity to handle these multi-modal data. The big models such as UNITER [27] and VisualBERT [91] deserve more attention in multi-modality MDS tasks. However, at present, there is little multi-modal research work based on MDS; this is a promising, but largely under-explored, area where more studies are expected.

8 CONCLUSION

In this article, we have presented the first comprehensive review of the most notable works to date on deep-learning-based multi-document summarization (MDS). We propose a taxonomy for organizing and clustering existing publications and devise the network design strategies based on the state-of-the-art methods. We also provide an overview of the existing multi-document objective functions, evaluation metrics, and datasets and discuss some of the most pressing open problems and promising future extensions in MDS research. We hope this survey provides readers with a comprehensive understanding of the key aspects of MDS tasks, clarifies the most notable advances, and sheds light on future studies.

REFERENCES

[1] Stergos Afantenos, Vangelis Karkaletsis, and Panagiotis Stamatopoulos. 2005. Summarization from medical documents: A survey. *Artificial Intelligence in Medicine* 33, 2, 157–177.

[2] Amanuel Alambo, Cori Lohstroh, Erik Madaus, Swati Padhee, Brandy Foster, Tanvi Banerjee, Krishnaprasad Thirunarayan, and Michael L. Raymer. 2020. Topic-centric unsupervised multi-document summarization of scientific and news articles. In *2020 IEEE International Conference on Big Data (BigData'20)*. 591–596.

[3] Mahfouth Alghamdi, Christoph Treude, and Markus Wagner. 2020. Human-like summaries from heterogeneous and time-windowed software development artefacts. In *Proceedings of the 6th International Conference of Parallel Problem Solving from Nature (PPSN'20)*. 329–342.

[4] Reinald Kim Amplayo and Mirella Lapata. 2021. Informative and controllable opinion summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL'21)*. 2662–2672.

[5] Stefanos Angelidis and Mirella Lapata. 2018. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*. 3675–3686.

[6] Diego Antognini and Boi Faltings. 2019. Learning to create sentence semantic relation graphs for multi-document summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization (NFiS'19)*. 32–41.

[7] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual question answering. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV'15)*. 2425–2433.

[8] Rachit Arora and Balaraman Ravindran. 2008. Latent Dirichlet allocation and singular value decomposition based multi-document summarization. In *Proceedings of the 2008 8th IEEE International Conference on Data Mining (ICDM'08)*. 713–718.

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR'15)*.

[10] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. 65–72.

[11] Elena Baralis, Luca Cagliero, Saima Jabeen, and Alessandro Fiori. 2012. Multi-document summarization exploiting frequent itemsets. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC'12)*. 782–786.

[12] Phyllis B. Baxendale. 1958. Machine-made index for technical literature - An experiment. *IBM Journal of Research and Development* 2, 4, 354–361.

[13] Patrice Bellot, Véronique Moriceau, Josiane Mothe, Eric SanJuan, and Xavier Tannier. 2016. INEX tweet contextualization task: Evaluation, results and lesson learned. *Information Processing and Management* 52, 5, 801–819.

[14] Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

[15] Lidong Bing, Piji Li, Yi Liao, Wai Lam, Weiwei Guo, and Rebecca J. Passonneau. 2015. Abstractive multi-document summarization via phrase selection and merging. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL'15)*. 1587–1597.

[16] Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2019. Unsupervised opinion summarization as copycat-review generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*. 5151–5169.

[17] Arthur Brazinskas, Mirella Lapata, and Ivan Titov. 2020. Few-shot learning for opinion summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*. 4119–4135.

[18] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS'20)*. 1877–1901.

[19] Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2017. Improving multi-document summarization via text classification. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI'17)*. 3053–3059.

[20] Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. 2015. Ranking with recursive neural networks and its application to multi-document summarization. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*. 2153–2159.

[21] Ziqiang Cao, Furu Wei, Sujian Li, Wenjie Li, Ming Zhou, and Houfeng Wang. 2015. Learning summary prior representation for extractive summarization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL'15)*. 829–833.

[22] Jaime G. Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International Conference on Research and Development in Information Retrieval (SIGIR'98)*. 335–336.

[23] Giuseppe Carenini, Raymond T. Ng, and Xiaodong Zhou. 2007. Summarizing email conversations with clue words. In *Proceedings of the 16th International Conference on World Wide Web (WWW'07)*. 91–100.

[24] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Maël Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner. 2005. The AMI meeting corpus: A pre-announcement. In *Machine Learning for Multimodal Interaction, 2nd International Workshop (MLMI'05)*. 28–39.

[25] Jiaao Chen and Diyi Yang. 2020. Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*. 4106–4118.

[26] Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. DialogSumm: A real-life scenario dialogue summarization dataset. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL'21)*. 5062–5074.

[27] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning. In *Proceedings of 16th European Conference on Computer Vision (ECCV'20)*. 104–120.

[28] Minhao Cheng, Jinfeng Yi, Pin-Yu Chen, Huan Zhang, and Cho-Jui Hsieh. 2020. Seq2Sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI'20)*. 3601–3608.

[29] Sangwoo Cho, Logan Lebanoff, Hassan Foroosh, and Fei Liu. 2019. Improving the similarity measure of determinantal point processes for extractive multi-document summarization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL'19)*. 1027–1038.

[30] Janara Christensen, Stephen Soderland, Oren Etzioni, et al. 2013. Towards coherent multi-document summarization. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL'13)*. 1163–1173.

[31] Eric Chu and Peter J. Liu. 2019. MeanSum: A neural model for unsupervised multi-document abstractive summarization. In *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*. 1223–1232.

[32] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems Workshop on Deep Learning (NIPS'14)*.

[33] Maximin Coavoux, Hady Elsahar, and Matthias Gallé. 2019. Unsupervised aspect-based multi-document abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization (NFiS'19)*. 42–47.

[34] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12, 2493–2537.

[35] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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 (NAACL-HLT'19)*. 4171–4186.

[36] Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL'14)*. 1370–1380.

[37] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Proceedings of the 33th Annual Conference on Neural Information Processing Systems (NeurIPS'19)*. 13042–13054.

[38] Cicero Dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of the International Conference on Computational Linguistics (COLING'14)*. 69–78.

[39] Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications* 165, 113679.

[40] Seppo Enarvi, Marilisa Amoia, Miguel Del-Agua Teba, Brian Delaney, Frank Diehl, Stefan Hahn, Kristina Harris, Liam McGrath, Yue Pan, Joel Pinto, et al. 2020. Generating medical reports from patient-doctor conversations using sequence-to-sequence models. In *Proceedings of the 1st Workshop on Natural Language Processing for Medical Conversations*. 22–30.

[41] Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research* 22, 457–479.

[42] Berna Erol, Dar-Shyang Lee, and Jonathan J. Hull. 2003. Multimodal summarization of meeting recordings. In *Proceedings of the 2003 IEEE International Conference on Multimedia and Expo (ICME'03)*. 25–28.

[43] Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics* 9, 391–409.

[44] Alexander R. Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R. Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL'19)*. 1074–1084.

[45] Xiachong Feng, Xiaocheng Feng, Bing Qin, Xinwei Geng, and Ting Liu. 2021. Dialogue discourse-aware graph convolutional networks for abstractive meeting summarization. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI'21)*. 3808–3814.

[46] Rafael Ferreira, Luciano de Souza Cabral, Frederico Freitas, Rafael Dueire Lins, Gabriel de França Silva, Steven J. Simske, and Luciano Favaro. 2014. A multi-document summarization system based on statistics and linguistic treatment. *Expert Systems with Applications* 41, 13, 5780–5787.

[47] Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10)*. 340–348.

[48] Yanjun Gao, Chen Sun, and Rebecca J. Passonneau. 2019. Automated pyramid summarization evaluation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL'19)*. 404–418.

[49] Yang Gao, Wei Zhao, and Steffen Eger. 2020. SUPERT: Towards new frontiers in unsupervised evaluation metrics for multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*. 1347–1354.

[50] Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond Ng, and Bita Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 1602–1613.

[51] Demian Gholipour Ghalandari, Chris Hokamp, Nghia The Pham, John Glover, and Georgiana Ifrim. 2020. A large-scale multi-document summarization dataset from the Wikipedia current events portal. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*. 1302–1308.

[52] Jade Goldstein, Vibhu O. Mittal, Jaime G. Carbonell, and Mark Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *Proceedings of the 2000 Conference of the North American Chapter of the Association for Computational Linguistics: Applied Natural Language Processing Conference (NAACL-ANLP'00)*. 91–98.

[53] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR'15)*.

[54] Travis R. Goodwin, Max E. Savery, and Dina Demner-Fushman. 2020. Flight of the PEGASUS? Comparing transformers on few-shot and zero-shot multi-document abstractive summarization. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING'20)*. 5640–5646.

[55] Quentin Grail, Julien Perez, and Eric Gaussier. 2021. 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 (EACL'21)*. 1792–1810.

[56] Max Grusky, Mor Naaman, and Yoav Artzi. 2018. 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 (NAACL-HLT'18)*. 708–719.

[57] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL'16)*. 1631–1640.

[58] Wenzhong Guo, Jianwen Wang, and Shiping Wang. 2019. Deep multimodal representation learning: A survey. *IEEE Access* 7, 63373–63394.

[59] Vishal Gupta and Gurpreet Singh Lehal. 2010. A survey of text summarization extractive techniques. *Journal of Emerging Technologies in Web Intelligence* 2, 3, 258–268.

[60] Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of the 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'09)*. 362–370.

[61] Majharul Haque, Suraiya Pervin, Zerina Begum, et al. 2013. Literature review of automatic multiple documents text summarization. *International Journal of Innovation and Applied Studies* 3, 1, 121–129.

[62] Tsutomu Hirao, Hidetaka Kamigaito, and Masaaki Nagata. 2018. Automatic pyramid evaluation exploiting Edu-based extractive reference summaries. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*. 4177–4186.

[63] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8, 1735–1780.

[64] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* 2, 5, 359–366.

[65] Ya-Han Hu, Yen-Liang Chen, and Hui-Ling Chou. 2017. Opinion mining from online hotel reviews–A text summarization approach. *Information Processing and Management* 53, 2, 436–449.

[66] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

[67] Jinbae Im, Moonki Kim, Hoyeop Lee, Hyunsouk Cho, and Sehee Chung. 2021. Self-supervised multimodal opinion summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP'21)*. 388–403.

[68] Sarthak Jain, Sarah Wiegreffe, Yuval Pinter, and Byron C. Wallace. 2020. Learning to faithfully rationalize by construction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*. 4459–4473.

[69] Anubhav Jangra, Adam Jatowt, Mohammad Hasanuzzaman, and Sriparna Saha. 2020. Text-image-video summary generation using joint integer linear programming. *Advances in Information Retrieval* 12036, 190.

[70] Anubhav Jangra, Sriparna Saha, Adam Jatowt, and Mohammad Hasanuzzaman. 2020. Multi-modal summary generation using multi-objective optimization. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'20)*. 1745–1748.

[71] Anubhav Jangra, Sriparna Saha, Adam Jatowt, and Mohammed Hasanuzzaman. 2021. Multi-modal supplementary-complementary summarization using multi-objective optimization. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'21)*. 818–828.

[72] Fred Jelinek, Robert L. Mercer, Lalit R. Bahl, and James K. Baker. 1977. Perplexity - A measure of the difficulty of speech recognition tasks. *Journal of the Acoustical Society of America* 62, S1, S63–S63.
[73] Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*. 2021–2031.
[74] Hanqi Jin, Tianming Wang, and Xiaojun Wan. 2020. Multi-granularity interaction network for extractive and abstractive multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*. 6244–6254.
[75] Anirudh Joshi, Namit Katariya, Xavier Amatriain, and Anitha Kannan. 2020. Dr. summarize: Global summarization of medical dialogue by exploiting local structures. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, (EMNLP'20)*. 3755–3763.
[76] Ambedkar Kanapala, Sukomal Pal, and Rajendra Pamula. 2019. Text summarization from legal documents: A survey. *Artificial Intelligence Review* 51, 3, 371–402.
[77] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 1746–1751.
[78] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR'17)*.
[79] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *Proceedings of the 8th International Conference on Learning Representations (ICLR'20)*.
[80] Jia Jin Koay, Alexander Roustai, Xiaojin Dai, Dillon Burns, Alec Kerrigan, and Fei Liu. 2020. How domain terminology affects meeting summarization performance. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING'20)*. 5689–5695.
[81] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS'12)*. 1106–1114.
[82] Sayali Kulkarni, Sheide Chammas, Wan Zhu, Fei Sha, and Eugene Ie. 2020. AQuaMuSe: Automatically generating datasets for query-based multi-document summarization. *arXiv preprint arXiv:2010.12694*.
[83] Sawan Kumar and Partha P. Talukdar. 2020. NILE: Natural language inference with faithful natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*. 8730–8742.
[84] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proceedings of the 8th International Conference on Learning Representations (ICLR'20)*.
[85] Logan Lebanoff, Kaiqiang Song, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019. Scoring sentence singletons and pairs for abstractive summarization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL'19)*. 2175–2189.
[86] Logan Lebanoff, Kaiqiang Song, and Fei Liu. 2018. Adapting the neural encoder-decoder framework from single to multi-document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*. 4131–4141.
[87] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 11, 2278–2324.
[88] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. 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 (ACL'20)*. 7871–7880.
[89] Haoran Li, Peng Yuan, Song Xu, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. Aspect-aware multimodal summarization for chinese e-commerce products. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI'20)*. 8188–8195.
[90] Haoran Li, Junnan Zhu, Cong Ma, Jiajun Zhang, and Chengqing Zong. 2017. Multi-modal summarization for asynchronous collection of text, image, audio and video. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*. 1092–1102.
[91] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.
[92] Piji Li, Lidong Bing, and Wai Lam. 2017. Reader-aware multi-document summarization: An enhanced model and the first dataset. In *Proceedings of the Workshop on New Frontiers in Summarization (NFiS'17)*. 91–99.
[93] Piji Li, Wai Lam, Lidong Bing, Weiwei Guo, and Hang Li. 2017. Cascaded attention based unsupervised information distillation for compressive summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*. 2081–2090.
[94] Wei Li, Xinyan Xiao, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du. 2020. Leveraging graph to improve abstractive multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*. 6232–6243.

[95] Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the Workshop of Text Summarization Branches Out*. 74–81.

[96] Chunyi Liu, Peng Wang, Jiang Xu, Zang Li, and Jieping Ye. 2019. Automatic dialogue summary generation for customer service. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD'19)*. 1957–1965.

[97] Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating Wikipedia by summarizing long sequences. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18)*.

[98] Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL'19)*. 5070–5081.

[99] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

[100] Annie Louis and Ani Nenkova. 2013. Automatically assessing machine summary content without a gold standard. *Computational Linguistics* 39, 2, 267–300.

[101] Yao Lu, Yue Dong, and Laurent Charlin. 2020. Multi-XScience: A large-scale dataset for extreme multi-document summarization of scientific articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*. 8068–8074.

[102] Wencan Luo, Fei Liu, Zitao Liu, and Diane J. Litman. 2016. Automatic summarization of student course feedback. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'16)*. 80–85.

[103] Congbo Ma, Wei Emma Zhang, Hu Wang, Shubham Gupta, and Mingyu Guo. 2021. Incorporating linguistic knowledge for abstractive multi-document summarization. *arXiv preprint arXiv:2109.11199*.

[104] Inderjeet Mani and Eric Bloedorn. 1997. Multi-document summarization by graph search and matching. In *Proceedings of the 14th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conference (AAAI'97)*. 622–628.

[105] Yuning Mao, Yanru Qu, Yiqing Xie, Xiang Ren, and Jiawei Han. 2020. Multi-document summarization with maximal marginal relevance-guided reinforcement learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*. 1737–1751.

[106] Paul W. McBurney and Collin McMillan. 2014. Automatic documentation generation via source code summarization of method context. In *Proceedings of the 22nd International Conference on Program Comprehension (ICPC'14)*. 279–290.

[107] Yishu Miao and Phil Blunsom. 2016. Language as a latent variable: Discrete generative models for sentence compression. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*. 319–328.

[108] Rada Mihalcea and Paul Tarau. 2005. A language independent algorithm for single and multiple document summarization. In *Proceedings of the 2nd International Joint Conference, Companion Volume to the Proceedings of Conference including Posters/Demos and Tutorial Abstracts (IJCNLP'05)*. 19–24.

[109] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NIPS'13)*. 3111–3119.

[110] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33nd International Conference on Machine Learning (ICML'16)*. 1928–1937.

[111] Sabine Molenaar, Lientje Maas, Verónica Burriel, Fabiano Dalpiaz, and Sjaak Brinkkemper. 2020. Medical dialogue summarization for automated reporting in healthcare. In *Proceedings of the International Conference on Advanced Information Systems Engineering (CAiSE Workshops'20)*. 76–88.

[112] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI'17)*. 3075–3081.

[113] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. 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 (NAACL-HLT'18)*. 1747–1759.

[114] Mir Tafseer Nayeem, Tanvir Ahmed Fuad, and Yllias Chali. 2018. Abstractive unsupervised multi-document summarization using paraphrastic sentence fusion. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING'18)*. 1191–1204.

[115] Preksha Nema, Mitesh M. Khapra, Anirban Laha, and Balaraman Ravindran. 2017. Diversity driven attention model for query-based abstractive summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*. 1063–1072.

[116] Ani Nenkova and Kathleen R. McKeown. 2012. A survey of text summarization techniques. In *Mining Text Data*. 43–76.

[117] Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing* 4, 2, 4.

[118] Ani Nenkova and Rebecca J. Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the 2004 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'04)*. 145–152.

[119] Jun-Ping Ng and Viktoria Abrecht. 2015. Better summarization evaluation with word embeddings for ROUGE. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP'15)*. 1925–1930.

[120] Ahmed Oussous, Fatima-Zahra Benjelloun, Ayoub Ait Lahcen, and Samir Belfkih. 2018. Big data technologies: A survey. *Journal of King Saud University-Computer and Information Sciences* 30, 4, 431–448.

[121] Shruti Palaskar, Jindrich Libovický, Spandana Gella, and Florian Metze. 2019. Multimodal abstractive summarization for how2 videos. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL'19)*. 6587–6596.

[122] Richard Yuanzhe Pang, Ádám Dániel Lelkes, Vinh Q. Tran, and Cong Yu. 2021. AgreeSum: Agreement-oriented multi-document summarization. In *Findings of the Association for Computational Linguistics (ACL-IJCNLP'21)*. 3377–3391.

[123] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*. 311–318.

[124] Rebecca J. Passonneau, Emily Chen, Weiwei Guo, and Dolores Perin. 2013. Automated pyramid scoring of summaries using distributional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL'13)*. 143–147.

[125] Ramakanth Pasunuru, Asli Celikyilmaz, Michel Galley, Chenyan Xiong, Yizhe Zhang, Mohit Bansal, and Jianfeng Gao. 2021. Data augmentation for abstractive query-focused multi-document summarization. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI'21)*. 13666–13674.

[126] Ramakanth Pasunuru, Mengwen Liu, Mohit Bansal, Sujith Ravi, and Markus Dreyer. 2021. Efficiently summarizing text and graph encodings of multi-document clusters. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'21)*. 4768–4779.

[127] Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18)*.

[128] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'18)*. 2227–2237.

[129] Maxime Peyrard. 2019. A simple theoretical model of importance for summarization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL'19)*. 1059–1073.

[130] Dragomir R. Radev. 2000. A common theory of information fusion from multiple text sources step one: Cross-document structure. In *Proceedings of the Workshop of the 1st Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL'00)*. 74–83.

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

[132] Dragomir R. Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara. 2013. The ACL anthology network corpus. *Language Resources and Evaluation* 47, 4, 919–944.

[133] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1, 8, 9.

[134] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21, 140:1–140:67.

[135] Paige Rodeghero, Collin McMillan, Paul W. McBurney, Nigel Bosch, and Sidney D'Mello. 2014. Improving automated source code summarization via an eye-tracking study of programmers. In *Proceedings of the 36th International Conference on Software Engineering (ICSE'14)*. 390–401.

[136] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1, 5, 206–215.

[137] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature* 323, 6088, 533–536.

[138] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic routing between capsules. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS'17)*. 3856–3866.

[139] Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*. 1073–1083.

[140] Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL'19)*. 2931–2951.

[141] Elaheh ShafieiBavani, Mohammad Ebrahimi, Raymond K. Wong, and Fang Chen. 2018. A graph-theoretic summary evaluation for rouge. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*. 762–767.

[142] Chintan Shah and Anjali Jivani. 2016. Literature study on multi-document text summarization techniques. In *Proceedings of the International Conference on Smart Trends for Information Technology and Computer Communications (SmartCom'16)*. 442–451.

[143] Ori Shapira, David Gabay, Yang Gao, Hadar Ronen, Ramakanth Pasunuru, Mohit Bansal, Yael Amsterdamer, and Ido Dagan. 2019. Crowdsourcing lightweight pyramids for manual summary evaluation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'19)*. 682–687.

[144] Nikhil S. Shirwandkar and Samidha Kulkarni. 2018. Extractive text summarization using deep learning. In *Proceedings of the 2018 4th International Conference on Computing Communication Control and Automation (ICCUBEA'18)*. 1–5.

[145] Abhishek Kumar Singh, Manish Gupta, and Vasudeva Varma. 2018. Unity in diversity: Learning distributed heterogeneous sentence representation for extractive summarization. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI'18)*. 5473–5480.

[146] Yan Song, Yuanhe Tian, Nan Wang, and Fei Xia. 2020. Summarizing medical conversations via identifying important utterances. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING'20)*. 717–729.

[147] Dan Su, Yan Xu, Tiezheng Yu, Farhad Bin Siddique, Elham J. Barezi, and Pascale Fung. 2020. CAiRE-COVID: A question answering and query-focused multi-document summarization system for COVID-19 scholarly information management. In *Proceedings of the 1st Workshop on NLP for COVID-19*.

[148] Simeng Sun and Ani Nenkova. 2019. The feasibility of embedding based automatic evaluation for single 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'19)*. 1216–1221.

[149] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR'14)*.

[150] Haihui Tan, Ziyu Lu, and Wenjie Li. 2017. Neural network based reinforcement learning for real-time pushing on text stream. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'17)*. 913–916.

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

[152] Amirsina Torfi, Rouzbeh A. Shirvani, Yaser Keneshloo, Nader Tavaf, and Edward A. Fox. 2020. Natural language processing advancements by deep learning: A survey. *arXiv preprint arXiv:2003.01200*.

[153] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL'16)*. 76–85.

[154] Jan Ulrich, Gabriel Murray, and Giuseppe Carenini. 2008. A publicly available annotated corpus for supervised email summarization. In *Proceedings of the 23th AAAI Conference on Artificial Intelligence in Enhanced Messaging Workshop (AAAI'08)*. 77–82.

[155] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS'17)*. 5998–6008.

[156] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS'15)*. 2692–2700.

[157] Tatiana Vodolazova, Elena Lloret, Rafael Muñoz, and Manuel Palomar. 2013. Extractive text summarization: Can we use the same techniques for any text? In *Proceedings of the 18th International Conference on Applications of Natural Language to Information Systems (NLDB'13)*. 164–175.

[158] Xiaojun Wan and Jianwu Yang. 2006. Improved affinity graph based multi-document summarization. In *Proceedings of the 2006 Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (NAACL'06)*. 336–347.

[159] Xiaojun Wan and Jianwu Yang. 2008. Multi-document summarization using cluster-based link analysis. In *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'08)*. 299–306.

[160] Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*. 6209–6219.

[161] Hu Wang, Qi Wu, and Chunhua Shen. 2020. Soft expert reward learning for vision-and-language navigation. In *Proceedings of the 16th European Conference on Computer Vision (ECCV'20)*. 126–141.

[162] Lu Wang and Wang Ling. 2016. Neural network-based abstract generation for opinions and arguments. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL'16)*. 47–57.

[163] Chien-Sheng Wu, Linqing Liu, Wenhao Liu, Pontus Stenetorp, and Caiming Xiong. 2021. Controllable abstractive dialogue summarization with sketch supervision. In *Findings of the Association for Computational Linguistics (ACL-IJCNLP'21)*. 5108–5122.

[164] Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. 2021. PRIMER: Pyramid-based masked sentence pretraining for multi-document summarization. *arXiv preprint arXiv:2110.08499*.

[165] Canwen Xu, Jiaxin Pei, Hongtao Wu, Yiyu Liu, and Chenliang Li. 2020. MATINF: A jointly labeled large-scale dataset for classification, question answering and summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*. 3586–3596.

[166] Runxin Xu, Jun Cao, Mingxuan Wang, Jiaze Chen, Hao Zhou, Ying Zeng, Yuping Wang, Li Chen, Xiang Yin, Xijin Zhang, Songcheng Jiang, Yuxuan Wang, and Lei Li. 2020. Xiaomingbot: A multilingual robot news reporter. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (ACL'20)*. 1–8.

[167] Min Yang, Chengming Li, Fei Sun, Zhou Zhao, Ying Shen, and Chenglin Wu. 2020. Be relevant, non-redundant, and timely: Deep reinforcement learning for real-time event summarization. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI'20)*. 9410–9417.

[168] Min Yang, Wei Zhao, Jianbo Ye, Zeyang Lei, Zhou Zhao, and Soufei Zhang. 2018. Investigating capsule networks with dynamic routing for text classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*. 3110–3119.

[169] Qian Yang, Rebecca J. Passonneau, and Gerard De Melo. 2016. PEAK: Pyramid evaluation via automated knowledge extraction. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*. 2673–2680.

[170] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R. Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Proceedings of the 33th Annual Conference on Neural Information Processing System (NeurIPS'19)*. 5754–5764.

[171] Kaichun Yao, Libo Zhang, Tiejian Luo, and Yanjun Wu. 2018. Deep reinforcement learning for extractive document summarization. *Neurocomputing* 284, 52–62.

[172] Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R. Fabbri, Irene Li, Dan Friedman, and Dragomir R. Radev. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *Proceedings of the 33th AAAI Conference on Artificial Intelligence (AAAI'19)*. 7386–7393.

[173] Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir R. Radev. 2017. Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL'17)*. 452–462.

[174] Wenpeng Yin and Yulong Pei. 2015. Optimizing sentence modeling and selection for document summarization. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'15)*. 1383–1389.

[175] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS'20)*. 17283–17297.

[176] David M. Zajic, Bonnie J. Dorr, and Jimmy Lin. 2008. Single-document and multi-document summarization techniques for email threads using sentence compression. *Information Processing and Management* 44, 4, 1600–1610.

[177] Jianmin Zhang, Jiwei Tan, and Xiaojun Wan. 2018. Adapting neural single-document summarization model for abstractive multi-document summarization: A pilot study. In *Proceedings of the 11th International Conference on Natural Language Generation (INLG'18)*. 381–390.

[178] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*. 11328–11339.

[179] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. 2018. Interpretable convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*. 8827–8836.

[180] Shiyue Zhang, Asli Celikyilmaz, Jianfeng Gao, and Mohit Bansal. 2021. 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 (ACL-IJCNLP'21)*. 6895–6909.

[181] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *Proceedings of the 8th International Conference on Learning Representations (ICLR'20)*.

[182] Wei Emma Zhang, Quan Z. Sheng, Ahoud Abdulrahmn F. Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology* 11, 3, 24:1–24:41.

[183] Wei Emma Zhang, Quan Z. Sheng, Adnan Mahmood, Dai Hoang Tran, Munazza Zaib, Salma Abdalla Hamad, Abdulwahab Aljubairy, Ahoud Abdulrahmn F. Alhazmi, Subhash Sagar, and Congbo Ma. 2020. The 10 research topics in the Internet of Things. In *Proceedings of 6th IEEE International Conference on Collaboration and Internet Computing (CIC'20)*. 34–43.

[184] Yong Zhang, Meng Joo Er, Rui Zhao, and Mahardhika Pratama. 2016. Multiview convolutional neural networks for multidocument extractive summarization. *IEEE Transactions on Cybernetics* 47, 10, 3230–3242.

[185] Mingjun Zhao, Shengli Yan, Bang Liu, Xinwang Zhong, Qian Hao, Haolan Chen, Di Niu, Bowei Long, and Weidong Guo. 2021. QBSUM: A large-scale query-based document summarization dataset from real-world applications. *Computer Speech and Language* 66, 101166.

[186] Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the Conference on Empirical Methods in Natural Language and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. 563–578.

[187] Xin Zheng, Aixin Sun, Jing Li, and Karthik Muthuswamy. 2019. 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'19)*. 3151–3160.

[188] Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*. 6197–6208.

[189] Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir R. Radev. 2021. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'21)*. 5905–5921.

[190] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*. 2921–2929.

[191] Chenguang Zhu, Yang Liu, Jie Mei, and Michael Zeng. 2021. MediaSum: A large-scale media interview dataset for dialogue summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'21)*. 5927–5934.

[192] Chenguang Zhu, Ruochen Xu, Michael Zeng, and Xuedong Huang. 2020. A hierarchical network for abstractive meeting summarization with cross-domain pretraining. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings (EMNLP'20)*. 194–203.

[193] Junnan Zhu, Haoran Li, Tianshang Liu, Yu Zhou, Jiajun Zhang, and Chengqing Zong. 2018. MSMO: Multimodal summarization with multimodal output. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*. 4154–4164.

[194] Markus Zopf. 2018. Estimating summary quality with pairwise preferences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'18)*. 1687–1696.

Received 9 November 2020; revised 12 February 2022; accepted 30 March 2022