Multi-document Summarization via Deep Learning Techniques: A Survey

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

Multi-document summarization (MDS) is an effective tool for information aggregation which generates an informative and concise summary from a cluster of topic-related documents. Our survey structurally overviews the recent deep learning based multi-document summarization models via a proposed taxonomy and it is the first of its kind. Particularly, we propose a novel mechanism to summarize the design strategies of neural networks and conduct a comprehensive summary of the state-of-the-art. We highlight the differences among various objective functions which are rarely discussed in the existing literature. Finally, we propose several future directions pertaining to this new and exciting development of the 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. 2020. Multi-document Summarization via Deep Learning Techniques: A Survey. 1, 1 (December 2020), 35 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

As technology advances in the rapidly developing era, the exponentially increasing of text data makes analyzing and understanding textual files a tedious work [51, 96]. From the readers' perspective, capturing the salient information from overwhelming documents is a labor-intensive and time-consuming task. The voluminous documents are urgently required to be processed more efficiently and the abundance of text data calls for text summarization techniques. Text summarization is one of the important tasks of natural language processing that automatically convert a text or a collection of texts within the same topic into a concise summary that contains key semantic information. The length of summaries is usually significantly less than the original text(s) [104]. The research on automatic text summarization has been attractive in the field of natural language processing [1, 87, 122] which can be beneficial for many downstream applications such as creating news digests, search, and report generation [100].

Authors' addresses: CONGBO MA, The University of Adelaide, congbo.ma@adelaide.edu.au; WEI EMMA ZHANG, The University of Adelaide, wei.e.zhang@adelaide.edu.au; MINGYU GUO, The University of Adelaide, mingyu.guo@adelaide.edu.au; HU WANG, The University of Adelaide, hu.wang@adelaide.edu.au; QUAN Z. SHENG, Macquarie University, michael.sheng@mq.edu.au.

XXXX-XXXX/2020/12-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

According to the number of input documents, text summarization can be cast into single document summarization and multi-document summarization. Single document summarization aims to form a summary from only one document while multi-document summarization aims at generating a short and informative summary across a set of topic-related documents. From the application perspective, single document summarization may not satisfy the requirement to produce comprehensive summaries, because it does not make good use of documents that are generated around the clock [103]. For content to be summarized, it is more comprehensive and accurate to generate a summary from multiple documents written at different times, covering different perspectives. From the technical point of view, multi-document summarization is more complicated and difficult to tackle than single document summarization [122]. This is because in the multi-document summarization task, there is more diverse and conflicting information among documents. The volume of documents is usually longer and the relations between documents are more complicated. In such large amount of documents, documents would inevitably be complement, overlapping and conflicting to each other [103]. In addition, excessively long input documents often lead to model degradation [56]. It is challenging for models to retain the most critical contents of complex input sequences, while generating the coherent, non-redundant, non-factual error and grammatically readable summaries. Therefore, multi-document summarization requires models to have stronger capabilities for analyzing the corpora, identifying and merging consistent information. Furthermore, multi-document summarization task is more computation-hungry, due to the increasing sizes of current datasets and language model parameters.

Multi-document summarization task enjoys a wide range of real world applications, including summarization on news [35], scientific publications [141], emails [20, 145], product reviews [40], lecture feedback [77, 78], Wikipedia articles generation [73], medical documents [1] and software project activities [2]. Recently, multi-document summarization technology has also received a great amount of attention in the industry. An intelligent multilingual news reporter bot named Xiaomingbot [136] was developed for news generation. This bot is able to summarize multiple news into one article and then translate it into multiple languages. Massive application requirements and rapidly growing online data promote the development of multi-document summarization. However, the majority of existing methods still generate summaries with manually crafted features [82, 130], such as sentence position features [11, 32], sentence length features [32], proper noun features [129], cue-phrase features [45], biased word features, sentence-to-sentence cohesion, sentence-to-centroid cohesion. The existing works using traditional algorithms can be divide into the following categories: term frequency-inverse document frequency (TF-IDF) based methods [10, 105], clustering based methods [42, 131], graph based methods [79, 130] and latent semantic analysis based methods [7, 46].

Deep learning has gained enormous attention in recent years due to its success in various domains, for instance, computer vision [62], natural language processing [29] and multi-modal [133]. Both industry and academia have been in a race to utilize deep learning to solve complex tasks due to its capability of capturing highly nonlinear relations of data. Recently, deep learning based models are applied in multi-document summarization [69, 74], which prospers the development of text summarization and enables models to achieve better performance. Comparing to the conventional approaches, deep learning based models reduce dependence on manual feature extraction drastically. This task attracts increasing attention in the natural language processing community and enjoys steady expansion ever since. The number of research publications on deep learning based multi-document summarization is increasing rapidly over the last five years – the statistics show the number of publications has 225% increase from 2017 to 2019. It provides strong evidence for the inevitable pervasiveness of deep learning in multi-document summarization research.

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 review articles are based on traditional algorithms instead of deep learning based methods [36, 47, 92, 116]. Therefore, we conduct this survey to embrace the knowledge of multi-document summarization. To the best of our knowledge, this is the first comprehensive survey in the direction of deep learning for multi-document summarization. This survey has been designed in a way such that it classifies the neural based multi-document summarization 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 panorama for researchers, practitioners and educators to quickly understand and step into the field of deep learning based multi-document summarization. The key contributions of this survey are three-folds:

- We propose a categorization scheme to organize the current works and provide a comprehensive review for deep learning based multi-document summarization techniques, including deep learning based models, objective functions, benchmark datasets and evaluation metrics.
- We review development movements and provide a systematic overview and summary of the state-of-the-art. We also summarize seven network design strategies based on our extensive studies on 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 the discussed matters.

Paper Selection. In this article, we select, summarize, discuss, and analyze 30 representative works. We used Google Scholar as the main search engine to discover related works. The high-quality papers are selected from top NLP and AI journals and conferences, include 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. In the following sections, this survey will cover various aspects of recent advanced deep learning based works in multi-document summarization. Section 2 gives an overview of multi-document summarization. Section 3 highlights network design strategies and offers a comprehensive review of deep learning based multi-document summarization techniques. This survey also summarizes objective functions in the literature (Section 4), evaluation metrics (Section 5), and available multi-document datasets (Section 6). Finally, section 7 discusses the future research directions for deep learning based multi-document summarization followed by the conclusion in Section 8.

2 OVERVIEW OF MULTI-DOCUMENT SUMMARIZATION

Before we dive into the details of this article, we start with an introduction of the definition, concepts and processing framework regarding multi-document summarization. The aim of multi-document summarization 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 sentences $\left\{s_{i,j} \mid j \in [1, M]\right\}$. $s_{i,j}$ refers to the j-th sentence in the i-th document. The standard summary Ref is called *golden summary* or

[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.

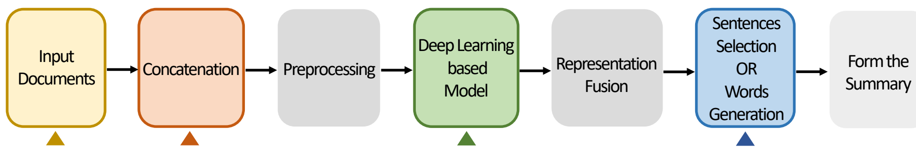

Fig. 1. The Processing Framework of Multi-document Summarization.

reference summary. Currently, most of the 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 multi-document summarization task, we summarize and illustrate the processing framework as shown in Figure 1. From the input document level, a significant difference between single document summarization and multi-document summarization is the number of input documents. Thus, detecting cross-document relations brings benefits for capturing salience information among documents. The first step is to select an appropriate concatenation approach for input documents. The second step is pre-processing these documents, such as segmenting sentences, tokenizing non-alphabetic characters and removing punctuations [118]. 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, through these five steps, multiple documents are transformed into concise and informative summaries.

Each of the highlighted steps (the ones with triangle mark) in Figure 1 involves various techniques. Accordingly, we classify the multi-document summarization task by input document types, concatenation methods, summarization construction types and deep learning based models. To provide a self-contained and comprehensive review, we also classify the research works from two other aspects: objective functions and evaluation metrics. Therefore, our proposed taxonomy categorizing the works from six aspects. Figure 2 depicts this categorization. In the rest of this section, we introduce the first three categorizations. The remaining three categories will be discussed in detail in Section 3, Section 4 and Section 5 respectively.

2.1 Input Document Types

In the multi-document summarization task, the types and length of input documents are diverse and they can be roughly divided into three groups:

- *Many short documents*. The length of each document is relatively short but the quantity of the input data is quite large. The typical representative of this type of data is product reviews [4]. Summarization on product reviews aims to generate a short and informative summary from numerous customer reviews of a product to reflect the public's overview.
- *Few long documents*. The length of each document is long but with a relatively small number of documents. For example, generating a brief summary from a group of news [35], constructing a Wikipedia style article from several web articles [73].
- *Hybrid documents*. One or some long documents with several short documents. Such as reader-aware multi-document summarization, which contains news with several readers' comments to this news [68]. Another example is forming a scientific summary from a long scientific paper with several short corresponding citations [141].

2.2 Concatenation Methods

A large number of input documents may contain contradiction, redundancy, and complementary information [103]. In order to deal with complex relations across input documents in the multi-document summarization task, we need concatenation methods to capture cross-document relations.

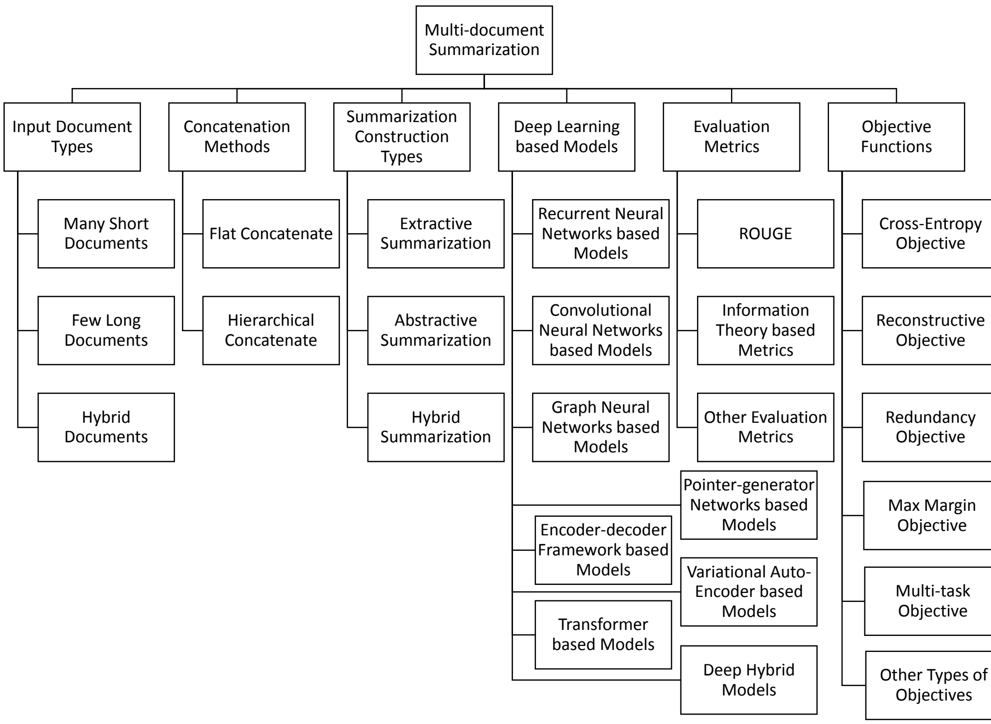

Fig. 2. Hierarchical Structure of This Survey.

There are two common types of methods to concatenate multiple input documents, namely: *flat concatenation* and *hierarchical concatenation* :

- *Flat Concatenation.* Flat concatenation is a simple yet powerful concatenation method. All input documents are spanned and are processed as a flat sequence. Therefore, with this concatenation method, the multi-document summary generation process can be seen as a single document summarization task. Inputting flat-concatenated documents requires models to have a strong ability of processing long-sequence. Consequently, how to enable models to handle long input sequences is crucial for multi-document summarization model designing.
- *Hierarchical Concatenation.* Different from flat concatenation, hierarchical concatenation methods are able to preserve cross-document relations. Taking advantage of hierarchical relations among documents instead of simply flat concatenating articles facilitates the model to obtain semantic-rich representation, which in turn improves the effectiveness of models. The input documents within a cluster describe a similar topic logically and semantically. Existing hierarchical concatenation methods either perform document-level condensing in a cluster separately [3] or process documents in word/sentence-level inside document cluster [5, 90, 132] to capture the cross-document relations. Figure 3 illustrates the two methods of hierarchical concatenation. Particularly, for the document-level concatenation methods, a condense model [3] is proposed to learn document-level representation separately in a cluster and these representation are fused in the subsequent processes. In Figure 3(b), sentences in the documents can be replaced by words. For the word/sentence-level concatenation methods, clustering algorithms and graph-based techniques are the most commonly used methods. More specifically, Mir et al. [90] perform sentence clustering for input multi-documents first allowing the model group related sentences. Then, the model selects sentences from

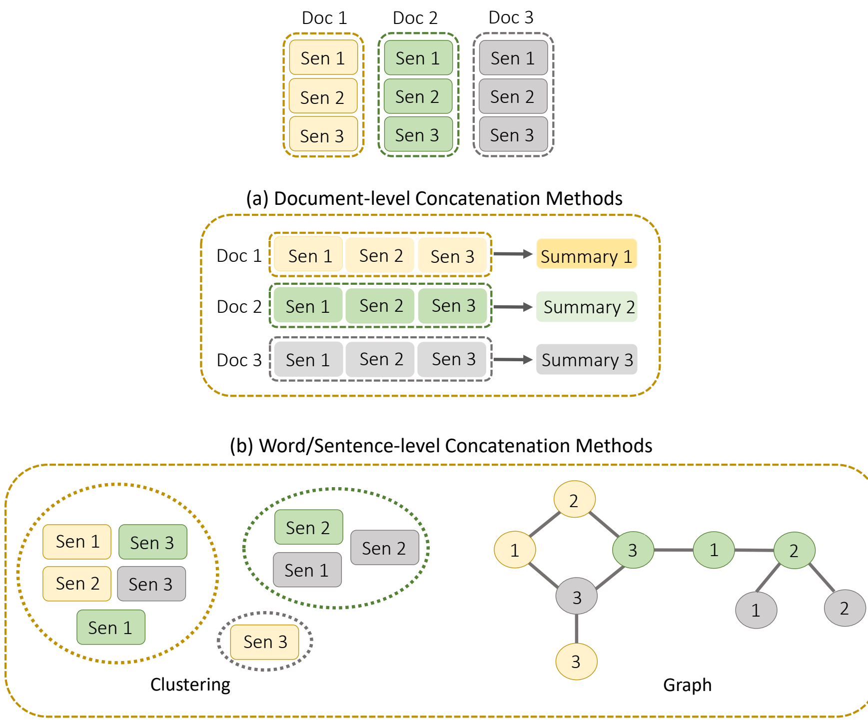

Fig. 3. The Methods of Hierarchical Concatenation.

diverse clusters, and at most one sentence will be selected from a cluster. By doing so, it will decrease redundancy and increase the information coverage for the generated summaries. The sentence relation graph is able to model hierarchical relations among multi-documents as well [5, 141, 142]. Most of the graph construction methods utilize sentences of vertexes and the edge between two sentences indicates their sentence-level relations [5]. Cosine similarity graph [32], approximate discourse graph [23] and personalized discourse graph [142] are the most commonly used methods recently for building sentence graph structures. The heterogeneous graph model [132] leverages words as intermediate nodes to construct a document-document, sentence-sentence and sentence-document hierarchical structure. In the structure, documents and sentences contain the same words connecting with each other through word nodes. However, many existing deep learning methods do not make full use of this hierarchical relationship in the document cluster [35, 73, 132].

2.3 Summarization Construction Types

Multi-document summarization seeks to compress a cluster of thematically-related documents into a short and informative summary. From the summarization construction types, the existing multi-document summarization methods can be grouped into three categories: *abstractive summarization*, *extractive summarization* and *hybrid summarization*. Figure 4 illustrates these three types.

- *Extractive Summarization.* Extractive summarization methods select salient snippets from the source documents to form informative summaries [88]. These methods generally contain two major components: *sentence ranking* and *sentence selection* [17]. Extractive summarization methods ensure the generated summaries are semantically similar to the original documents. However, these methods face several challenges: i) how to select the most "meaningful"

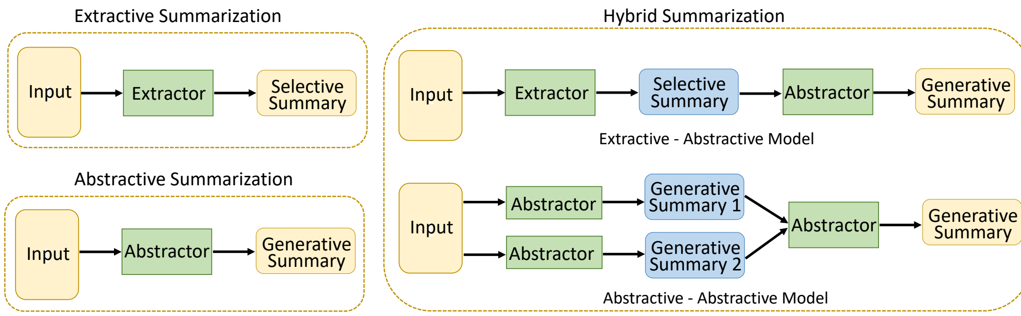

Fig. 4. Summarization Construction Types for Multi-document Summarization.

content, ii) how to improve the coherence and flexibility of generated summaries, and iii) how to reduce the redundant information among selective sentences.

- *Abstractive Summarization.* Abstractive summarization methods aim at presenting the main information of input documents, while automatically generating summaries are both succinct and coherent. This cluster of methods allows models to generate new words and sentences from a corpus pool [100]. Comparing to extractive summarization methods, the processing of abstractive summarization is more similar to the human-written ones. The challenge of abstractive summarization is that it requires more sophisticated natural language understanding and generation techniques, such as paraphrasing and sentence fusion techniques.
- *Hybrid Summarization.* In the multi-document summarization task, the input documents have complex textual relations. It is especially difficult to capture these documents semantically, as well as to fuse disparate features. In such a situation, hybrid models are proposed to combine the advantages of the aforementioned extractive and abstractive methods to process the input texts. The common used hybrid summarization models usually process data in two stages: extractive-abstractive and abstractive-abstractive (show in Figure 4 (right)). These two-stage models try to gather important information from source documents with extractive or abstractive methods at the first stage, by doing which it can reduce the length of input documents significantly. Later on, the processed important texts from the prior step are fed into an abstractive model to form final summaries.

3 DEEP LEARNING BASED MULTI-DOCUMENT SUMMARIZATION METHODS

Deep neural network models show a strong capability to fit given data in a variety of research fields, such as computer vision [62] and natural language process [29]. For the multi-document summarization task, deep neural networks also gain considerably better performance than traditional methods. Comparing to the traditional methods, due to the high non-linearity and strong fitting abilities of neural based models, deep neural methods can effectively process large-scale documents and distill informative summaries. In this section, we first introduce our novel way that generalizes seven neural network design strategies (Section 3.1). Then we present the state-of-the-art deep neural network based multi-document summarization models according to the main neural network architecture they adopt (Section 3.1 - 3.9).

3.1 Architecture Design Strategies

Architecture design strategies play a critical role in deep learning based models. Most of the existing neural networks are applied in the multi-document summarization task with variants or ensembled versions. We generalize the reviewed network architectures and summarize them into seven types. These seven type structures can also be used as basic structures and can stack on each other to obtain more diverse design strategies. Figure 5 illustrates these network design strategies. In this figure, deep neural models are boxed in green dotted line, which can be flexibly substituted by other backbone networks. The blue solid line boxes indicate the neural embeddings processed by neural networks or heuristic-designed approaches. It can be sentence/document representation or other types of representation. We summarize the network design strategies according to how to generate or fuse semantic-rich and syntactic-rich representation to improve multi-document summarization model performance. The explanation of each sub-figure is listed as follows:

- *Naive Networks.* In Figure 5(a), DNN based models play the role as a feature extractor. Multiple concatenated documents are input through DNN based models to extract word-level, sentence-level or document-level representation. The representation is used for the downstream summary generation or sentence selection. *Naive networks* represent the most naive model which lays the foundation for other strategies.
- *Ensemble Networks.* Ensemble methods leverage multiple learning algorithms to obtain better performance than individual algorithms. In order to capture semantic-rich and syntactic-rich representation, *Ensemble networks* (Figure 5(b)) feed input documents to multiple paths with different network structures or operations. Later on, these representations are fused to enhance model expression capability. The majority vote or average can be used to determine the final solution.
- *Auxiliary Task Networks. Auxiliary task networks* (Figure 5(c)) employ different tasks in the summarization models. For example, text classification, text reconstruction or other auxiliary tasks serve as complementary representation learners to obtain advanced features. Meanwhile, *Auxiliary task networks* also provide researchers a solution to further utilize adequate data from other tasks. In this strategy, the weights of different branches are shared.
- *Reconstruction Networks.* Labeling multi-document summarization data is a time-consuming and laborious work. *Reconstruction networks* (Figure 5(d)) optimize models from unsupervised learning paradigm, which allow summarization models overcome the limitation of insufficient labeled golden summaries. By using such a paradigm, the generated summaries are able to be constrained in the natural language domain in a good manner.
- *Fusion Networks. Fusion networks* (Figure 5(e)) fuse representation generated from neural networks and hand-crafted features, such as sentence position embeddings or other types of embeddings. Those hand-crafted features contain adequate prior knowledge which facilitates the optimization of summarization models.
- *Graph Neural Networks.* Figure 5(f) shows the general structure of *Graph neural networks* leveraged by multi-document summarization task. Constructing graph structure based on the source documents, including word, sentence or document level information, are conducted in *Graph neural networks*. This strategy captures cross-document relations which is crucial and beneficial for multi-document model training.
- *Hierarchical Networks.* Figure 5(g) shows the structure of *Hierarchical networks*. Multiple documents are also concatenated as inputs to feed into the first DNN based model to capture the low-level representation. Then 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 Networks based Models

Recurrent Neural Network (RNN) [110] is a class of neural networks that excels in modeling sequential data. It intends to capture sequential relations and syntactic/semantic information from word sequences. Inside RNN models, the neurons are connected through hidden layers. Different from other neural network structures, the inputs of each RNN neuron are not only from word or sentence embedding but also from the output of the previous hidden state. Despite the fact that RNN are powerful, vanilla RNN models often encounter gradient explosion or vanishing issues. Thus, a

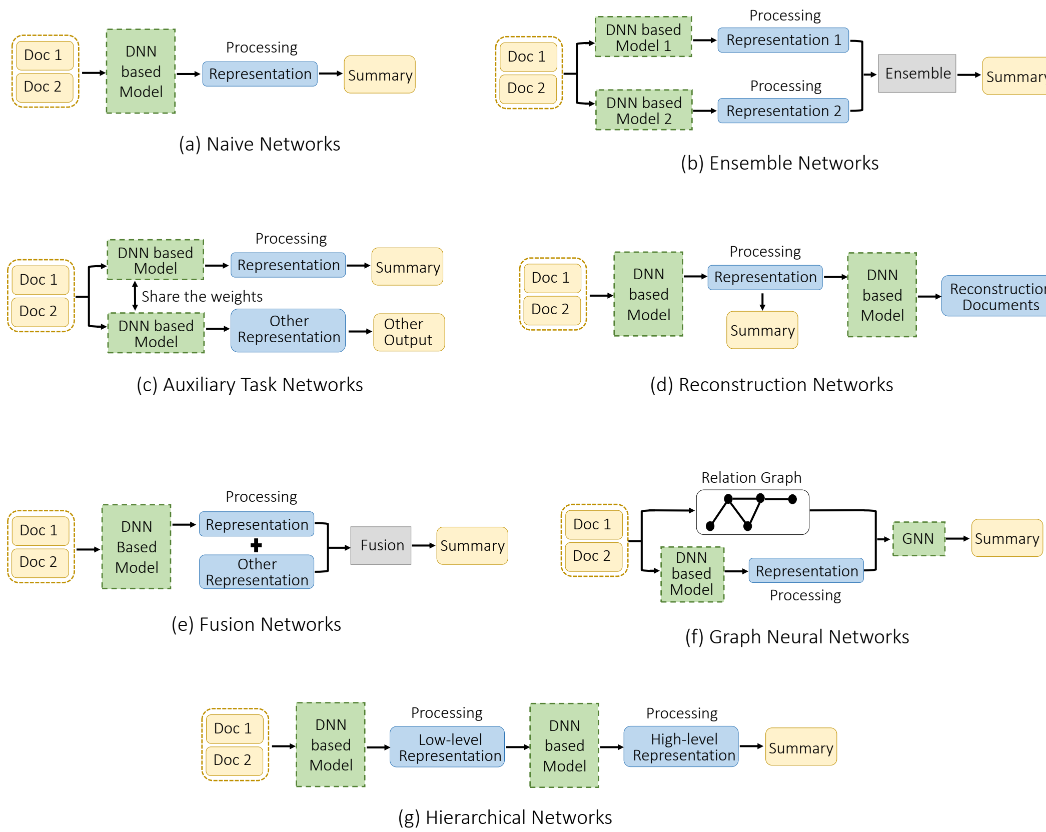

Fig. 5. Network Design Strategies.

large number of RNN-variants are proposed to conquer these problems. The most prevalent ones are Long Short-Term Memory (LSTM) [50], Gated Recurrent Unit (GRU) [25] and Bi-directional Long Short-Term Memory (Bi-LSTM) [52]. The *DNN based Model* in Figure 5 can be replaced by these RNN based models to design networks.

RNN-based models are adopted by multi-document summarization task since 2015. Cao et al. [17] proposed a RNN-based model termed *Ranking framework upon Recursive Neural Networks (R2N2)*, which leverages manually extracted word-level 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. Li et al. [69] developed a RNN-based framework to estimate the salience information from documents in an unsupervised manner. This framework is employed for information distillation to extract salience information vectors from the input sentences. In this process, 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.

The premise of multi-document summarization assumes the set of documents belongs to the same topic. However, within the documents set, documents may belong to the same/different subtopics and the salience of sentences are different concerning different subtopics. Zheng et al. [155] used hierarchical RNN structure to extract sentence, document and topic embeddings from the original documents. In this subtopic-driven multi-document summarization model *STDS*, the reader's comments as auxiliary documents assist to extract more concise and semantic information.

Reinald et al. [3] claimed that the summaries generated by *extractive-abstractive* models are less informative, inaccurate and these models can not easily generate specific summaries based on the preferences. According to these drawbacks, they proposed a two-stage *Condense-Abstract (CA)* framework, viewing the opinion summarization as an instance of multi-source transduction, to distill salient information from source documents. For each input document, the condense model leverages a Bi-LSTM auto-encoder to learn document-level and word-level representations. The hidden states in both directions of Bi-LSTM are concatenated as word-level representation. The document representation consists of the first and the last word encodings. The abstract model fuses multi-source representation and generates an opinion summary with a simple LSTM decoder combined with a vanilla attention mechanism [8] and a copy mechanism [127].

Since paired document summarization datasets are rare and hard to obtain, Chu et al. [24] proposed an unsupervised end-to-end abstractive summarization architecture called *MeanSum*. This LSTM-based model formalizes the product or business reviews summary problem into two individual close-loops. The first close-loop represents each of the input reviews onto an encoded domain and reconstructs those embeddings back to the language domain without requiring golden summary signals. Similarly, in the second close-loop, the paper brings out a concept named *combined review representation* that is averaging the encoded reviews to form more comprehensive reviews. Then the same LSTM-based encoder and decoder in the first close-loop are leveraged to reconstruct the combined review representation projected back to the encoded domain, during which the summary is generated from the weights-shared decoder.

Inspired by MeanSum, Maximin et al. [26] utilized a two-layer standard LSTM to construct sentence representation for aspect-based multi-document abstractive summarization. The sentence representation is clustered by a supervised aspect-based classifier. For each cluster, the model computes individual representation to generate a sentence summary. The authors mentioned that the clustering strategy empowers the model to reward the diversity of reviews and handle contradictory ones. Different from aspect-based multi-document summarization, Arthur et al. [13] introduced a GRU-based encoder-decoder architecture to minimize the diversity of opinions reflecting the dominant views while generating the multi-reviews summaries.

3.3 Convolutional Neural Networks Based Models

Convolutional neural networks (CNNs) [67] achieves excellent results in the computer vision tasks. The convolution operation scans through the embeddings of word/sentence. Through convolution kernels, the important information of the input data objects can be extracted by convolutional neural networks. Using pooling operation on features at intervals can get different levels of features from simple to complex. Convolutional neural networks have been proved to be effective for various NLP tasks in recent years [31, 58]. They are able to process natural language after sentences/words vectorization. Most of the CNN-based multi-document summarization models utilize CNNs for semantic and syntactic feature representation. In network design strategies (Figure 5), CNN based models could replace the *DNN based Model*.

A simple way to make use of CNNs in multi-document summarization is by sliding multiple filters with different window sizes over the input documents for semantic representation. Cao et al. [18] proposed a hybrid CNN-based model *PriorSum* to capture the 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 [27] to fetch document independent features that are more informative than using standard CNNs. This paper also nominates to combine manually-extracted document dependent features with the aforementioned document independent features to provide representation in a better manner for sentence ranking. Similarly, *HNet* [119] utilizes distinct CNN filters and max-over-time-pooling to generate salient feature representation for the downstream processes. Cho et al. [22] also used different filter sizes in *DPP-Combined* model to extract low-level feature.

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

Models	Nodes	Edges	Edge Weights	GNN Methods
HeterDoc-SumGraph[132]	word, sentence, document	word-sentence, word-document	TF-IDF	Graph Attention Networks
Graph-based Neural MDS [142]	sentence	sentence-sentence	Personalized Discourse Graph	Graph Convolutional Networks
SemSentSum[5]	sentence	sentence-sentence	Cosine Similarity Graph Edge Removal Method	Graph Convolutional Networks
ScisummNet[141]	sentence	sentence-sentence	Cosine Similarity Graph	Graph Convolutional Networks

With regard to the summarization task, different people tend to have different summaries of the same documents due to different background knowledge and understanding. In this case, a better summary should be generated by exchanging one's opinion and negotiating among the summarizers. Inspired by the general idea of imitating human summarizers' behaviors, Zhang et al. [152] suggested a *MV-CNN* model which ensembles three individual models to incorporate the idea of multi-view learning and CNNs to improve the performance of multi-document summarization. In this work, three CNNs with dual-convolutional layers utilize multiple filters with different window sizes to extract distinct saliency scores of the sentences. By doing so, the MV-CNN model can better capture semantic and syntactic information of the documents. After saliency scores being computed, *consensus principle* and a two-level *complementary principle* of multi-view learning are leveraged to calculate the final sentence scores.

Cao et al. [16] brought out a *TCSum* model incorporating an auxiliary text classification sub-task into the multi-document summarization bringing in more supervision signals. The text classification model utilizes a CNN descriptor to project the documents onto the distributed representation and to classify input documents into different categories. The summarization model shares the projected sentence embedding generated by the classification model. Then, the TCSum model chooses the corresponding category-based transformation matrices according to classification results to transform the sentence embedding into the summary embedding. Finally, the sentence selection can be predicted by the sentence salience score which is calculated based on the cosine similarity between summary embedding and sentence embedding.

Yin et al. [143] presented an unsupervised CNN-based model termed *Novel Neural Language Model (NNLM)* to extract sentence representation and diminish the redundancy of sentence selection. The framework of NNLM contains only one convolution layer and a max-pooling layer. Within the NNLM, both element-wise averaging sentence representation and context words representation are used to predict the next word. In terms of aspect-based opinion summarization, Stefanos et al. [4] leveraged a CNN model to encode the product reviews which contain a set of segments for opinion polarity.

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

3.4 Graph Neural Networks Based Models

Convolutional neural networks have been successfully applied to many computer vision tasks to extract distinguished image features from the Euclidean space. However, convolutional neural methods will present difficulties when processing non-Euclidean data. Natural language data consist of vocabularies and phrases with strong relations and they can be better represented with graphs rather than in sequential orders. Graph neural network (GNN) is an ideal architecture for natural language processing since it is able to model the strong relations between entities semantically and syntactically. Graph convolution networks and graph attention networks are the most commonly adopted GNNs because of their efficiency and simplicity for integration with other neural networks. Figure 5 (f) shows the general structure of the graph neural networks based models. The models firstly build a relation graph based on the input documents, where nodes can be words, sentences or documents and edges capture the similarity among them. At the same time, the input documents are fed into a DNN based model to generate embeddings in different levels. The graph neural networks are then built on the top to capture the salient contextual information. Table 1 represents the current graph neural models for multi-document summarization with details of nodes, edges, edge weights and applied GNN methods.

Yasunage et al. [142] developed a graph convolution network (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 [60] to capture the 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 cosine similarity graph [32], approximate discourse graph [23] and the proposed personalized discourse graph. Both of the sentence relation graph and sentence embeddings extracted by a sentence-level RNN are fed into graph convolution networks 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 et al. [5] proposed a GCN based model named *SemSentSum* which constructs a graph based on the sentence relations. Different from [142], this work leverages the external universal embeddings, which are 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 similarity. If the weight of the edge is lower than the threshold then remove this edge. This sentence relation graph and sentence embeddings are fed into a GCN [60] to generate saliency estimation for extractive summaries.

Yasunage et al. [141] also designed a GCN based model for summarizing scientific papers. The proposed *ScisummNet* model not only employs the abstract of source scientific paper but also employs the text spans from the papers that cite the original paper. The total number of citations is also incorporated in the model as an authority feature. Cosine similarity graph is applied to form the sentence relation graph. Then, graph convolution networks are adopted to predict sentence salience estimation from the sentence relation graph, the authority scores and the sentence embeddings.

The existing graph neural networks based models are mainly focused on the relationship between sentences and do not fully consider the relations among words, sentences and documents. To capture semantic relations between words, sentences and documents, Wang et al.[132] proposed a heterogeneous graph attention networks [126] for extractive multi-document summarization, which is termed as *HeterDoc-Sum Graph*. This heterogeneous graph structure includes word nodes, sentence nodes and document nodes. Sentence nodes and document nodes are connected according to the contained word nodes. Therefore, the word nodes act as an intermediate bridge to connect the sentence nodes and the document nodes. In this case, the document-document relations, sentence-sentence relations and sentence-document relations can be better established through word nodes. TF-IDF values are utilized as the weights of word-sentence and word-document edges. Then the node representation of these three levels are passed into the graph attention networks for model updating. In each updating iteration, bidirectional updating processes of both word-sentence and word-document are performed to better aggregate cross-level semantic knowledge.

3.5 Pointer-generator Networks Based Models

Pointer-generator networks [113] is proposed to overcome the problems of factual errors and high redundancy in the summarization task. This network is inspired by pointer network [128], copynet [44], forced-attention sentence compression [81] and coverage mechanism from machine translation [123]. In the network, the probability distribution of vocabularies and input sequences can be obtained through the Sequence-to-sequence (Seq2Seq) model predictions and the pointer networks respectively. The pointer-generator networks combine the aforementioned two models to obtain a united probability distribution allowing vocabularies to select from source texts or generated by machines. Additionally, the coverage mechanism prevents pointer-generator networks from choosing the same phrases all the time, which may incur the same predictions.

Alexander et al. [35] proposed an end-to-end *Hierarchical MMR-Attention Pointer-generator (Hi-MAP)* model to incorporate pointer-generator networks and *Maximal Marginal Relevance (MMR)* [19] for multi-document abstractive summarization. To better obtain important sentences and filter the redundant information in the summary, the Hi-MAP model improves pointer-generator networks by modifying its attention weights. More accurately, the MMR scores are multiplied to the original attention weights. MMR method is designed to select a set of salience sentences from source documents by considering both *importance* and *redundancy indexes* [19]. The redundancy controls the selected sentence has as little overlap as possible with the existing summary. The MMR model adds a new sentence to the objective summary each time based on the importance score and redundancy score until the summary length reaches a certain threshold.

Similarly, the MMR approach was implemented as well by *PG-MMR* [66] model to identify salience source sentences from multi-document inputs. But the calculation of MMR scores is different between Hi-MAP and PG-MMR. ROUGE-L Recall and ROUGE-L Precision are employed to calculate the importance score and redundancy score in the proposed PG-MMR model. To overcome the scarcity of multi-document datasets, the PG-MMR model leverages a support vector regression model that is pretrained on a single-document dataset to empower the model to recognize the important content. This support vector regression model also calculates the score of each input sentence by considering four factors: sentence length, sentence relative/absolute positions, sentence-document similarities, and sentence qualities obtained by a pointer-generator network. Sentences with top-K scores are fed into another pointer-generator network to generate a concise sentence. Then by following the principle of the MMR standard, the model decides whether to add the newly generated sentence to the summary. Each time after a sentence is generated, the model updates scores for all candidate sentences. The process repeats until a summary length threshold is reached.

3.6 Encoder-decoder Based Models

Encoder-decoder [8] is a neural network based design pattern. It is one of the successful architectures for sequential text data processing. This architecture is partitioned into two parts, namely encoder and decoder. For multi-document summarization, the encoder embeds source documents into the hidden representations, i.e., word representation, sentence representation and document representation. Then, the representation containing compressed semantic and syntactic information is passed to the decoder to generate the target summaries. Figure 6 shows the process of encoder-decoder framework. RNN-based encoder-decoder structures are commonly used [13, 24, 69]. In the last two years, transformer-based encoder and decoder structures gain increasing heed in the multi-document summarization community [56, 71, 74].

Generally, in encoder-decoder based multi-document summarization models, the encoder structure can be either sentence encoder or document encoder. When it serves as the sentence encoder, the inputs are the word embeddings of each sentence, and the outputs are the sentence embeddings. When it is the document encoder, the inputs are sentence embeddings for the given documents, and the outputs are the document embeddings. Later on, the decoder processes the latent embeddings to synthesize local and global semantic/syntactic information to produce the final summaries.

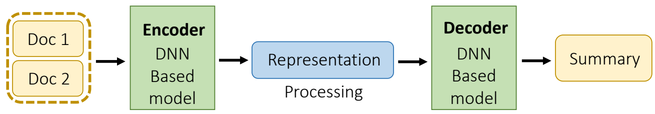

Fig. 6. The Process of Encoder-Decoder Framework.

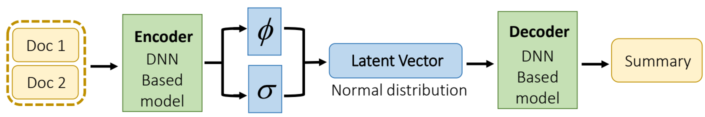

Fig. 7. The Process of Variational Auto-Encoder Framework.

Recently, a big surge has occurred to use pre-trained language model encoders, such as BERT [28], GPT-2 [107], GPT-3 [15]. The models are fine-tuned with randomly initialized decoders in an end-to-end fashion since transfer learning can assist the model training process [71]. Thus, more "meaningful" representation with prior knowledge will be introduced by leveraging pre-trained language models.

3.7 Variational Auto-Encoder Based Models

The goal of auto-encoder (AE) [48] is to minimize the reconstructive distance between the inputs and the predicted outputs to achieve the purpose of dimension reduction. Variational Auto-Encoder (VAE) [30, 59] is an application of encoder-decoder framework for reconstruction [24, 155] and is also a commonly used generative model. Figure 7 shows the framework of variational auto-encoder based model. The basic principle of VAE is to transform the latent bottle neck vector into a normal distributed vector to better capture the low dimensional manifold for continuous generation. Some existing methods introduce VAE into multi-document summarization task to capture the manifold of latent features [13, 70].

Li et al. [68] proposed a VAE based model for reader-aware multi-document summarization which takes news and reader comments into consideration simultaneously. The variational auto-encoder is employed to catch the complex manifold of data in the latent space. At the same time, latent semantics are modeled by the variational bottleneck and thus condensed features are extracted for further unsupervised data reconstruction framework an alignment mechanism in the next step. The subsequent work [70] also follows the paradigm of variational auto-encoder framework for unsupervised multi-document summarization task. The main idea is to jointly model the news sentence and the comment sentence by the proposed weighted alignment mechanism. Arthur et al. [14] proposed an end-to-end hierarchical variational auto-encoder model for unsupervised abstractive reviews summarization. This model aims at minimizing the novelty of product reviews which reflects the dominant opinions.

3.8 Transformer Based Models

As discussed, CNN based models are not good at processing sequential data as RNN based models. However, RNN based models are not easy to parallel computing as the current states in RNN models highly depend on the results from the previous steps. Additionally, RNNs are difficult to process long sequences since former knowledge will fade away during the learning process. Adopting *Transformer* model [125] is one solution to solve the above problems. The Transformer is based on the self-attention mechanism and has natural advantages for parallelization and retains long-range dependencies. The Transformer model has achieved promising results in multi-document summarization tasks [56, 71, 73, 74]. The *DNN based Model* in Figure 5 can also be replaced by these Transformer based models to design networks.

Liu et al. [73] introduced Transformer into multi-document summarization task. The goal is to generate a Wikipedia article from a given topic and a set of references. This paper argues that the encoder-decoder based sequence transduction model is difficult to deal with long input documents. Targeting this issue, the model firstly selects a series of top-K tokens and feeds them into a Transformer based decoder-only sequence transduction to generate the Wikipedia articles. More specifically, the transformer decoder only architecture combines the result from the extractive stage and golden summary into a sentence for training.

Yang et al. [74] proposed a *Hierarchical Transformer model (HT)* with an inter-paragraph and graph-informed attention mechanism allowing the model encodes multiple input documents hierarchically instead of simple flat-concatenation. The decoding process follows a vanilla Transformer architecture [125] and incorporates a length penalty for short summaries and low fluency. Additionally, a logistic regression model is employed to select top-K paragraphs and these paragraphs are pumped 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.

Based on the HT model, Li et al. [71] proposed an end-to-end Transformer based model *GraphSum* which leverages graph structure to capture the cross-document relations. The hierarchical encoder leverages several token-level Transformer encoding layers to extract token embeddings and these embeddings are pumped into the high-level Transformer based graph encoding layers. 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 on the summarization model.

To obtain rich semantic representation from different granularity, Jin et al. [56] proposed a Transformer based multi-granularity interaction network *MGSum* and unified the extractive and abstractive multi-document summarization. Words, sentences and documents are three granularity of semantic units that are connected by a granularity hierarchical relation graph. In the same granularity, a self-attention mechanism is utilized 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* employed a fusion gate to integrate and update the semantic representation. Additionally, a spare attention mechanism [153] is leveraged to guarantee the summary generator focusing on the important information.

3.9 Deep Hybrid Models

Many neural based models can be integrated to formalize a more powerful and expressive model because of the promising results. In this section, we summarize the existing deep hybrid models that have proven to be effective in the multi-document summarization task.

CNN + LSTM + Capaule networks. Cho et al. [22] proposed a hybrid model based on the determinantal point processes for semantically measuring the sentence similarity. A convolutional layer slides over the pairwise sentences with filters of different sizes to extract low-level features. Capsule networks [111, 137] are employed to identify redundant information by transforming the spatial and orientational relationships for high-level representation. The authors also used LSTM to reconstruct the pairwise sentences and add reconstruction loss to the final objective function.

CNN + Bi-LSTM + MLP. Abhishek et al. [119] proposed an extractive multi-document summarization framework that incorporates document-dependent and document-independent information. In this model, a CNN with different filters captures the phrase-level representation. Full binary trees formed with these salient representation are fed to the recommended Bi-LSTM tree indexer to enable better generalization abilities. A multi-layer perceptron (MLP) with the 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.

Table 2. Deep Learning based Methods.

DNN Architeture	Works	Construction Types			Concatenation Methods		Supervision Types	
		Extractive	Abstractive	Hybrid	Flat	Hierarchical	Supervised	Unsupervised
RNN	[24]		✓		✓			✓
	[146]		✓		✓		✓	
	[155]	✓				✓		✓
	[90]		✓			✓		✓
	[17]	✓			✓		✓	
	[3]			✓		✓	✓	
	[69]		✓		✓			✓
	[134]		✓		✓		✓	
	[26]		✓		✓			✓
CNN	[152]	✓			✓		✓	
	[16]	✓			✓		✓	
	[143]	✓			✓			✓
	[18]	✓			✓		✓	
	[4]	✓			✓		✓	
GNN	[142]	✓				✓	✓	
	[5]	✓				✓	✓	
	[141]	✓				✓	✓	
	[132]	✓				✓	✓	
PG	[66]		✓		✓		✓	
	[35]		✓		✓		✓	
VAE	[68]		✓		✓			✓
	[14]		✓		✓			✓
	[70]		✓		✓			✓
Transformer	[74]		✓			✓	✓	
	[56]	✓	✓			✓	✓	
	[71]		✓			✓	✓	
	[73]			✓	✓			✓
Deep hybrid model	[22]	✓			✓		✓	
	[65]		✓		✓		✓	
	[119]	✓			✓		✓	

Pointer-generator networks + Transformer. In the process of generating a summary, in addition to compressing or rewriting a single sentence, it is also necessary to consider the information fusion of multiple sentences, especially the sentence pairs. Through analyzing the human-written summary from three popular summarization datasets, Logan et al. [65] found that the majority of summary sentences are generated by fusing one or two source sentences. Therefore, they proposed a two-stage summarization method that considers the semantic compatibility of the sentence pairs. This method joint scores single sentence and sentence pair to filter representative single sentence and sentence pair from the original documents. Then sentences or sentence pair with high scores are compressed and rewritten to get a summary which leverages the pointer-generator networks. This paper uses a Transformer based model to encode both single sentence and sentence pairs indiscriminately to obtain the deep contextual representation of words and sequences.

Discussion. In this section, we reviewed the state-of-the-art works of deep learning based multi-document summarization models according to the neural networks applied. Table 2 summarizes the reviewed works by applying three classification schemes: *construction type*, *concatenation methods* and *supervision types*. According to the presence or absence of golden summaries, summarization models can be divided into supervised learning models and unsupervised learning models. The input document type of models depends on the datasets they used. In the case of a shortage of multi-document summarization datasets, auto-encoder based models can be utilized to optimize models in an unsupervised manner. Furthermore, these models adopt the text itself as supervision signals which constrains the generated summaries in the natural language domain. Transformer based models are the most commonly used models in the last two year because they overcome the limitations of CNN's fixed-size receptive field and RNN's unparalleled.

4 OBJECTIVE FUNCTIONS

In this section, we will take a closer look at different objective functions adopted by various multi-document summarization models. In deep learning based differentiable summarization models, loss functions are used to calculate the similarity between the model outputs and the ground-truth to further guide the model optimization process through backpropagation. Objective functions are crucial for the development of multi-document summarization research, which gives us the motivation to survey different proposed objective functions. Previously, cross-entropy (CE) based loss functions are often accepted to measure the distance between the model generated summaries and the golden summaries. However, by only adopting cross-entropy loss alone may not lead the model to a generalizable minima. Several objective variants are thus proposed, such as reconstructive objective that introduces unsupervised signals, redundancy objective that minimizes the overlap information in the summary max-margin objective, and multi-task objective that seeks supervision signals from other tasks. These objectives provide researchers various constraints from different angles for better optimization.

4.1 Cross-Entropy Objective

Cross-Entropy usually acts as an objective function to measure the distance between two distributions. Many existing multi-document summarization works adopt it to measure the difference between the distributions of generated summaries and the golden summaries [17, 22, 132, 141, 146, 152]. Formally, 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 salience score that is usually calculated by Recall-Oriented Understudy for Gisting Evaluation (ROUGE), which we will discuss in Section 5. $\hat{\mathbf{y}}_\mathbf{i}$ is the predicted estimation which usually learned from the deep learning based models. The *HT* model [74] uses cross-entropy loss to minimize the distance between the paragraph score vectors $\hat{\mathbf{y}}_\mathbf{i}$ and the ground truth vectors $\mathbf{y_i}$. More specifically, ROUGE-2 recall is adopted to compute the ground truth score between the selected paragraphs and golden summary. Paragraph score vectors are calculated by an LSTM-based model. Yasunaga et al. [142] combined graph convolution networks with document summarization task to fetch better representation. The normalized average of ROUGE-1 and ROUGE-2 scores are treated as the ground truth in the cross-entropy objective. In the model, the cluster embeddings are obtained by averaging over several document embeddings. Finally, the loss can be calculated by considering both of the cluster embeddings and the representation of a single sentence. An end-to-end training model *SemSentSum* [5] also adopts cross-entropy loss as its objective. Different from the common methods in the summarization task, the ROUGE-1 F_1 score for each sentence is calculated rather than recall values to alleviate long sentences selection tendency of the model.

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:

$$L_{Rec} = D[\mathbf{x_i}, \phi'(\phi(\mathbf{x_i}; \theta); \theta')], \tag{2}$$

where $\mathbf{x_i}$ represents the input vector. ϕ and ϕ' represent the encoder and decoder with θ and θ' as their parameters respectively. $D(.)$ stands for a measuring function to calculate the distance between source documents and their reconstructive outputs, usually L_* distance ($_*$ stands for 0, 1, 2, infinity, p ...) are used. Chu et al. [24] utilized a reconstructive loss to constrain the generated text into the natural language domain. This model reconstructs the reviews in a token-by-token manner. Moreover, the 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 then the summary is decoded to the reconstructed reviews to form another reconstructive close-loop. An unsupervised learning loss is designed by [69] to reconstruct the condensed output vectors to the original input sentence vectors with L_2 distance. Specially, the paper further constrains the condensed output vector with a L_1 regularizer to ensure its sparsity. Similarly, Zheng et al. [155] adopted a bidirectional GRU encoder-decoder framework to reconstruct both news sentences and comment sentences in a word sequence manner. Liu et al. [73] used reconstruction within its abstractive stage. A two-stage strategy is used to alleviate the problem brought in by long input documents. Both input and output sequences are concatenated to predict the next token for the training of 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. [68] introduced three individual reconstructive losses to consider both news reconstruction and comments reconstruction respectively along with a variational auto-encoder lower bound. Similarly, Li et al. [70] optimized the aforementioned three reconstructive objectives as well. Moreover, a reconstruction loss for VAEs-A and VAEs-Zero has been further introduced. Bravzinskas et al. [13] utilized Variational auto-encoder to generate the latent vectors of given reviews in a better manner. Then, 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, the model is encouraged to maximize its information coverage. Formally,

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

where $Sim(.)$ 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. Redundancy Objective is often treated as an auxiliary objective function that is used in combination with other loss functions. Li et al. [69] penalized phrases pairs with similar meanings as well to eliminate the redundancy. Nayeem et al. [91] utilized redundancy objective to avoid generating repetitive phrases and this objective constrains a sentence only appears once while maximizing the scores of important phrases. Zheng et al. [155] adopted redundancy loss function to measure the overlaps between subtopics. Intuitively, the smaller overlaps between subtopics, the less redundancy exists in the output domain. Besides this loss, this work also proposes another contrastive loss to urge pair of topic representation to be close to each other within the same proposed subtopic. Yin et al. [143] proposed a redundancy objective to estimate the diversity between different sentences. This paper not only considers the pair-wise similarities, but also takes the probability of each sentence to be selected termed as prestige into account. The intuition behind the method is that two sentences will bring different effects to the result even when they are similar. In this case, the higher the prestige, the heavier the penalty would be.

4.4 Max Margin Objective

Max Margin Objectives (MMO) are also used in multi-document summarization task to empower the model to learn more distinctive 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 as the parameters of model function $f(.)$, γ is the margin threshold. The max-margin objective aims to force that function $f(\mathbf{x_i};\theta)$ and function $f(\mathbf{x_j};\theta)$ be separated by a predefined margin γ. Cao et al. [16] proposed two objective functions for classification task and summarization task respectively. The one for summarization task is a max margin objective to constrain a pair of randomly sampled sentences with different saliency scores: the higher score should be larger than the other one more than a marginal threshold. Two max margin losses are proposed in [156]: a margin-based triplet loss that encourages the model to pull the golden summaries semantically closer to the original documents than the machine generated summaries; a pair-wise margin loss based on the intuition that the farther apart the paired candidates in the ROUGE score rankings, the greater margin they hold.

4.5 Multi-task Objective

Supervision signals from multi-document summarization objectives may not be strong enough for representation learners, 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 multi-document summarization task, L_{Other} is the loss function of an auxiliary task. Angelidis et al. [4] incorporated multi-task classification loss since they claim that the reconstruction signal is weak. The work assumes that the aspect-relevant words not only provide a reasonable basis for model aspect reconstruct, but also a good indicator for product domain. Similarly, multi-task classification is introduced by [16]. Two models are maintained: one is a text classification model and the other is a summarization model. In the first model, CNN networks are used to classify text categories and the cross-entropy loss is used as the objective function. The summarization model and the text classification model share the convolutional parameters and pooling operation, so it is equivalent to the shared document vector representation. Maximin et al. [26] jointly optimized the model from a language modeling objective and other two multi-task supervised classification losses, which are polarity loss and aspect loss, to fetch better results.

4.6 Other Types of Objectives

In addition to the above-mentioned objectives, there are many other types of objectives. Cao et al. [18] proposed to use ROUGE-2 to calculate the sentence saliency scores and the model tries to estimate this saliency with linear regression. Yin et al. [143] suggested to sum up the square of the prestige vectors calculated by the PageRank algorithm to identify the importance of the sentences. Zhang et al. [152] proposed an objective function by ensembling the individual scores obtained from multiple CNN models. Besides the cross-entropy loss, a consensus objective is adopted in this model to minimize the disagreement between every two classifiers. Amplay et al. [3] utilized two objectives in abstract module: the first one is to optimize the generation probability distribution by maximizing the likelihood; the second objective is to constrain the output of the model to be close to its golden summary in the encoding space, as well as being distant from the random sampled negative summaries. Chu et al. [24] designed similarity objective which shares the encoder and decoder weights within the auto-encoder module. In the summarization module, the average cosine distance is used to indicate 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 which is the average of the cells states and hidden states to constrain the generated summaries semantically close to reviews.

Discussion. In multi-document summarization, cross-entropy is the most commonly adopted objective function. It bridges the predicted candidate summaries and the golden summaries in an intuitive manner by treating the ground-truth summaries as strong supervision signals. Reconstruction objectives offer a view from the unsupervised learning perspective. By using these objectives, model optimization could be conducted with the documents themselves if the manual annotation is scarce. The works which adopt multi-task objectives explicitly define multiple auxiliary tasks to assist the main summarization task for better generalization. To constrain models semantically, the Redundancy objective takes text overlapping information into account. Max margin objectives require a certain score outperform the other score by a predefined threshold.

5 EXISTING EVALUATION METRICS

Evaluation metrics intend to measure the effectiveness of a given method in an objective way. In this case, well-defined and effective evaluation metrics are crucial to multi-document summarization research. Currently, a large amount of evaluation metrics are developed for the multi-document summarization task. We classify the existing evaluation metrics in three categories and will discuss each category in detail: (i) ROUGE: the most commonly used evaluation metrics in the summarization community, (ii) Information theory based metrics: a simple evaluation criterion based on information theory, and (iii) Other evaluation metrics: these evaluation metrics have not been widely used in multi-document summarization research so far.

5.1 ROUGE

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [72] is a collection of evaluation indicators and is one of the most essential metrics for many natural language processing tasks, including machine translation, single document summarization and multi-document summarization. ROUGE obtains prediction/ground-truth similarity score through comparing automatically generated summaries with a set of corresponding human-written references word-wisely. As such an important evaluation metric, ROUGE has many variants to measure the candidate abstracts in a variety of ways [72]. The commonly used ones are ROUGE-N and ROUGE-L.

5.1.1 ROUGE-N. *ROUGE-N* [72] stands for ROUGE with N-gram Co-Occurrence Statistics. It measures a n-gram recall between reference summaries and their corresponding candidate summaries. Formally, ROUGE-N can be calculated as:

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

where Ref is the reference summaries and n represents n-gram. $Count_{match}(gram_n)$ represents the maximum number of n-grams in the reference summaries and the corresponding candidates. The numerator of ROUGE-N is the number of n-grams owned by both the reference summaries and the automatically generated ones, while the denominator is the total number of n-grams occurring in the golden summary. The denominator could be set to the number of candidate summary n-grams as well to measure the precision. However, ROUGE-N mainly focuses on quantifying recall, so precision is not calculated here.

ROUGE-1 and ROUGE-2 are the special cases of ROUGE-N that are usually chosen as the best practices. The number 1 and 2 represent unigram and bigram, respectively. ROUGE-1, ROUGE-2 or ROUGE-N are adopted by most of the research works [4, 16, 143]. For example, the referenced golden summary Ref (usually written by humans) is *“A man was on the bridge”*; and the machine-generated summary Sum is *“A man was found on the bridge”*. In this case, the golden summary has six unigrams and five bigrams, while the reference abstract has seven unigrams and six bigrams respectively. Thus the $ROUGE_1(Sum, Ref) = \frac{6}{6} = 1.0$ and $ROUGE_2(Sum, Ref) = \frac{4}{5} = 0.8$.

5.1.2 ROUGE-L. *ROUGE-L* [72] acquires its name from the first letter of *Longest Common Subsequence*, indicating the longest common subsequence algorithm is used as the essential part of ROUGE-L to count the longest matching vocabularies. 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

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

In the above equations, LCS(.) represents the longest common subsequence function, Ref and Sum are golden summary and machine generated summary. ROUGE-L is termed as 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}. However, in some circumstances, by setting β to a big number, only R_{lcs} is considered. The use of ROUGE-L enables us to measure the similarity of the two text sequences in the sentence-level. ROUGE-L also has the advantage of automatically decide the n-gram without extra manual works, since the calculation of LCS empowers the model to count grams adaptively. Specifically, when ROUGE-L is applied to the summary-level matching, the LCS scores between a reference summary sentence and each of the corresponding candidate sentence are calculated to fetch R_{lcs} and P_{lcs}.

5.1.3 ROUGE-W. *ROUGE-W* [72] is the F-measure of co-occurrence precision and recall rate, which is known as a weighted longest common subsequence. Given two sentences with different vocabulary orders presenting the same ROUGE-L score, usually they should be treated differently. More specifically, if one of the sentences has more consecutive co-occurred words with the golden sentence than the other one, this sentence should be a better choice. Thus, ROUGE-W is proposed to weighted the consecutive matches for the better measurement of semantic similarities between two texts under this circumstance.

5.1.4 ROUGE-S & ROUGE-SU. *ROUGE-S* [72] stands for ROUGE with Skip-Bigram Co-Occurrence Statistics that allows the bigram to skip arbitrary words. Skip-Bigrams are pairs of words but allow gaps. Take *"A man was there"* for instance, skip-bigrams are *"A man", "A was", "A there", "man was", "man there", "was there"*. Because Skip-Bigram can connect a pair of words that are far from each other. Similar to the calculation of ROUGE-L, for ROUGE-S score, skip-bigram-based F-measure is computed. By using ROUGE-S, it enables evaluation metrics to measure long-distance word-pair relations. However, if no constraint of the skipping words is set, a large amount of spurious matches will be produced. Thus, the maximum skip distance is a hyper-parameter to be tuned.

ROUGE-SU [72] refers to ROUGE with Skip-bigram plus Unigram-based co-occurrence statistics. As an extension of ROUGE-S, the unigram information is further added to ROUGE-SU to conquer the potential zero ROUGE-S score issue. ROUGE-SU is also able to be obtained from ROUGE-S by adding a begin-of-sentence token at the starting point of both references and candidates.

5.1.5 ROUGE for Multiple References. In terms of ROUGE metric for multiple golden summaries, the Jackknifing procedure, which is similar to K-fold-validation, is introduced [72]. Detailly, take ROUGE-N for instance, M golden summaries are given, the M best scores could be computed from sets composed by $M - 1$ reference summaries. Then, the final ROUGE-N is the average of the aforementioned M scores. This procedure can also be applied to ROUGE-L, ROUGE-W and ROUGE-S.

5.1.6 ROUGE-WE. ROUGE-WE stands for ROUGE evaluation with Word Embeddings. ROUGE measures the distances of summaries literally. Different from the exact word-to-word matching in ROUGE, Ng et al. [95] introduced a soft-matching evaluation metric named *ROUGE-WE*. ROUGE-WE is proposed to further extend ROUGE to measure the pair-wise summary distances in word embeddings space. Having few common vocabularies in the literal space of a text pair may result in failure by adopting ROUGE. However, these texts may incur high similarity in the embedding space. So it can be successfully detected when ROUGE-WE is adopted. More formally, the similarity function is defined as:

$$f_{WE}(w_1, w_2) = \begin{cases} 0, \text{ if } v_1 \text{ or } v_2 \text{ is OOV} \\ v_1 \cdot v_2, \text{ otherwise} \end{cases} \tag{10}$$

where w_1 and w_2 are the comparing words. v_1 and v_2 are the embeddings of w_1 and w_2 respectively. *OOV* (out-of-vocabulary) represents a scenario that no embedded vector with respect to a word w. In recent years, more ROUGE-based evaluation models have been proposed to compare golden summaries and machine-generated summaries not just according to the literal similarity, but also consider on semantic similarity [115, 149, 154].

5.2 Information Theory based Metrics

ROUGE is generally based on word-level matching, which inevitably possesses problems in scenarios where sentences with high ROUGE scores show low semantic similarities to the golden summaries. Differently, Peyrard [102] proposed three simple but rigorous evaluation criteria from the aspect of information theory, including (i) *Redundancy*, (ii) *Relevance*, and (iii) *Informativeness*. Then a final indicator termed (iv) *Importance* is calculated from the aforementioned three aspects.

5.2.1 Redundancy. Intuitively, a summary should contain a large amount of information which is measured according to the Shannon's entropy. More specifically, for a summary *Sum* presented by its semantic units frequency distribution $\mathbb{P}_{Sum}$, the Shannon's entropy $H(Sum)$ is:

$$H(Sum) = -\sum_{\mathbf{x_i}} \mathbb{P}_{Sum}(\mathbf{x_i}) \cdot \log\left(\mathbb{P}_{Sum}(\mathbf{x_i})\right), \tag{11}$$

where $\mathbf{x_i}$ represents a semantic unit. If each semantic unit appears only once within the summary, $H(Sum)$ is maximized. In this case, *Redundancy* is defined as:

$$Red(Sum) = H_{\max} - H(Sum), \tag{12}$$

where $H_{\max}$ is a constant, which is independent to the maximized $H(Sum)$. Therefore, the simplified version of *Redundancy* calculation is $Red(Sum) = -H(Sum)$.

5.2.2 Relevance. Relevance is introduced to evaluate whether the generated sentence contains important information related to the original corpus that should result in a minimum loss of information. The candidate summary is presented by *Sum* and the source documents are presented by D. Their distributions are represented by $\mathbb{P}_{Sum}$ and $\mathbb{P}_D$, respectively. *Relevance* is measured through the negative of the cross-entropy of two distributions $\mathbb{P}_{Sum}$ and $\mathbb{P}_D$, which represents $Rel(Sum, D) = -CE(Sum, D)$. More formally:

$$Rel(Sum, D) = \sum_{\mathbf{x_i}} \mathbb{P}_{Sum}(\mathbf{x_i}) \cdot \log\left(\mathbb{P}_D(\mathbf{x_i})\right), \tag{13}$$

From the equation, we can see that $Rel(Sum, D)$ is illustrated as the average novelty if we expect D but observe *Sum*.

5.2.3 Informativeness. When the abstract is generated, it is necessary not only to ensure that the result is expressed in the document but also to include as much new information as possible to avoid repeated information that the readers already know. Thus, a new notion *BK* of is involved to present background knowledge. The *Informativeness* is calculated by the cross-entropy between *Sum* and *BK* as $Inf(Sum, BK) = CE(Sum, BK)$. Formally, we have:

$$Inf(Sum, BK) = -\sum_{\mathbf{x_i}} \mathbb{P}_{Sum}(\mathbf{x_i}) \cdot \log\left(\mathbb{P}_{BK}(\mathbf{x_i})\right). \tag{14}$$

Different from the measurement of *Relevance*, in the calculation of *Informativeness* we expect this cross-entropy to be high. In the training process, modeling *BK* at the same time of generating summaries enable the model to produce not only a better summary but also based on a better knowledge background.

5.2.4 Importance. A good summary should have low *Redundancy* and high *Relevance* as well as high *Informativeness*. According to the paper [102], $\mathbb{P}_{\frac{D}{BK}}$ encodes the relative importance of different semantic units. In this case, a candidate summary should approximate $\mathbb{P}_{\frac{D}{BK}}$. If θ_I is the summary scoring function, to unify all aforementioned three indicators, we have:

$$\theta_I(Sum, D, BK) \equiv -\,\mathrm{Red}(Sum) + \alpha\,\mathrm{Rel}(Sum, D) + \beta\,\mathrm{Inf}(Sum, BK). \tag{15}$$

The minimization of *Redundancy* as well as maximization of *Relevance* and *Informativeness* are conveyed by maximizing θ_I. It can be used as a guided summary scoring function for model training in the future, and a better summary can be generated by comprehensively considering various aspects of previously mentioned metrics.

5.3 Other Evaluation Metrics

Besides *ROUGE*-based [72] and information theory-based [102] metrics, there are plenty of existing evaluation metrics for multi-document summarization. These evaluation metrics have received less attention than ROUGE so far. We hope this section will give researchers and practitioners a holistic view on the evaluation metrics in this field.

Perplexity [54] is used to evaluate the quality of the language model by calculating the negative of log probability of the appearance of a word. In this case, a low perplexity on a test dataset is a strong indicator of the high quality of a summary grammatically because it measures the probability of words following their previous words.

Pyramid is a semantic-based evaluation metric. The abstract sentence is manually divided into several Summarization Content Units (SCUs) and each of them represents a core concept. After sorting SCUs in the order of importance to form the *Pyramid* [93], the quality of automatic summarization is evaluated by calculating the number and importance of SCUs included in the document [94]. Each SCU is formed from a single word to a sentence. Intuitively, more important SCUs exist at higher levels of pyramid. Although pyramid shows its strong correlation with human judgments, it requires professional annotations to match and evaluate the SCUs in generated summaries and their ground truth summaries. Some recent works are focusing on the construction of *Pyramid* [38, 49, 99, 117, 138].

Responsiveness [76] measures the content selection and linguistic quality of summaries by directly rating scores. Additionally, the assessments are calculated without reference to model summaries. For text content evaluation, *Pyramid* and *Responsiveness* are commonly used.

SUPERT proposed by Gao et al. [39] is an unsupervised multi-document summarization evaluation metrics to measure the semantic similarity between the pseudo reference summary and the machine generated summary. The *SUPERT* no longer requires golden summaries which avoids human annotations. Contextualized embeddings and soft token alignment techniques are leveraged to select salient information from the input documents to evaluate the quality of the summaries.

Preferences-based metric [159] is a pairwise sentence preferences-based evaluation model. It does not depend on the golden summaries or Pyramid Summarization Content Units annotations. The basic idea is to ask annotators about their pair-wise preference rather than writing complex golden summaries. Compared to the traditional reference summaries-based evaluation models, pairwise preferences are much easier to obtain.

Discussion. Although there are many evaluation metrics for multi-document summarization, the indicators of the ROUGE series are generally accepted by the natural language processing community. Almost all the research works utilize ROUGE for evaluation, while other evaluation indicators are just for assistance. 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 used in machine translation task or other natural language processing tasks, such as BLEU [98] and METEOR [9], could be potentially used in multi-document summarization task.

Table 3. Comparison of Different Datasets. In the table, "Ave", "Summ", "Len" and "#" present average, summary, length and numbers; "Docs" and "sents" mean documents and sentences; is ; "Bus" and "Rev" represent business and reviews .

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 1579360 / 38144 / 38205	1 summ / cluster	139.4 tokens/ summ	Wikipedia
Multi-news	-	train / val / test 44972 / 5622 / 5622 2-10 docs / cluster	1 summ / cluster	263.66 words / summ 9.97 sents / summ 262 tokens / summ	News
Opinosis	51	6457 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: 10695 / 1337 / 1337 rev: 1038184 / 129856 / 129840	-	-	Customer reviews
Reader Aware	45	450 docs	4 summ / cluster	100 words	News
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

6 DATASETS

Comparing to single document summarization tasks, large-scale datasets of multi-document summarization are relatively scarce. In this section, we present our investigation on the existing multi-document summarization datasets.

DUC & TAC. DUC[9] stands for Document Understanding Conference. DUC provides official text summarization competitions each year from 2001 to 2007 to promote the research of this field. DUC changes its name to Text Analysis Conference (TAC)[10] in 2008. In our paper, the DUC datasets refer to the data collected from 2001 to 2007; the TAC datasets refer to the dataset after 2008. Both DUC and TAC datasets are from the news domains including various topics such as Politics, Natural Disaster and Biography. Different topic clusters are endorsed to classify the corpus into several subsets. For each cluster, multiple golden summaries are provided by expertise. Word level or byte-level limitations of each golden summary are given as well. Nevertheless, as shown in Table 3, DUC and TAC datasets provide small datasets for model evaluation that only include hundreds of news documents and human-annotated summaries. In addition,

the first sentence in a news item usually is information-rich that renders bias in the news datasets. Thus, it fails to reflect various natural documents in our daily lives. These two datasets are on a

[9]http://duc.nist.gov/
[10]http://www.nist.gov/tac/

relatively small scale and are not ideal for large-scale deep neural multi-document summarization model training and evaluation.

Oposum. Oposum [4] introduces a new dataset for model training and evaluation. It collects multiple reviews of six product domains from Amazon. The dataset not only contains multiple reviews and corresponding summaries but also products' domain information and polarity information. The latter information could be used as the auxiliary supervision signals.

Wikisum. Different from most of the existing datasets for extractive multi-document summarization, Wikisum [73] targets abstractive multi-document summarization, which generates summarization automatically by machines rather than selects phrases from the original documents. The Wikisum dataset treats generating English Wikipedia as a supervised abstractive summarization task. More specifically, a Wikipedia theme with a set of non-Wikipedia reference documents is pumped into neural networks. The model tries to fit real Wikipedia articles acting as supervision signals. These non-Wikipedia references are crawled from documents cited by Wikipedia, as well as the top-10 Google search results with the titles of documents. However, some of the URLs are not available and part of them are identical to each other. For these problems, Liu et al. [74] cleaned the dataset and further deleted the duplicated examples. Under the circumstances, we report the statistical results from [74].

Multi-news. As we stated previously, the DUC and TAC datasets are in a relatively small scale that may not be ideal for the training and evaluation of the deep neural models. Alternatively, in the news domain, Multi-news [35] is a large-scale dataset containing various topics for multi-document summarization. In Multi-news, news articles and human-written summaries are all from the Web[11]. The Multi-news dataset contains trace-back links to the original documents and also includes 56,216 article-summary pairs. Moreover, in the paper, the authors compared multi-news dataset with prior datasets in terms of coverage, density and compression. The results show that the multi-news dataset has various arrangement styles of sequences.

Opinosis. Opinosis Dataset [37] contains reviews of 51 topic clusters collected from TripAdvisor, Amazon and Edmunds. In terms of each topic, approximately 100 sentences on average are provided and the reviews are fetched from different sources. For each cluster, five professional written golden summaries are provided for the model training and evaluation.

Rotten Tomatoes Dataset. Rotten tomatoes dataset [134] consists of the collected reviews of 3,731 movies from rotten tomato website[12]. Among them, the authors selected 2,458 movies, 536 movies and 737 movies for training, validation and testing. The reviews contain both professional critics and user comments. For each movie, one sentence summary is created by professional editors.

Yelp. Chu et al. [24] proposed a dataset named Yelp based on the Yelp Dataset Challenge. Yelp dataset includes multiple customer reviews with five-star ratings. Furthermore, the authors also provided 100 manual-written summaries for model evaluation using Amazon Mechanical Turk (AMT), within which every eight input reviews are summarized into one golden summary.

Reader-Aware. Reader-Aware [68] is another news domain dataset on multi-document summarization. The documents in the Reader-Aware dataset are gathered from 45 different topic clusters and six predefined classes. The paper points out that other than the news itself, comments from readers toward this news also play important roles while summarizing news. The comment information provides new perspectives for the multi-document summarization task.

SciSumm. SciSumm dataset [141] is a large manually-annotated corpus for scientific documents summarization. The summarization model not only needs to summarize a main scientific document which is called *reference paper (RP)* but also to summarize multiple sentences from the paper that cites this reference paper. In the SciSumm dataset, 1,000 most cited papers from ACL Anthology

[11] http://newser.com
[12] http://rottentomatoes.com

Network [106] are treated as reference papers. Moreover, for each reference paper, 15 citation sentences (on average) are also provided after cleaning. Multi-document summarization models can seek adequate information from these citation sentences. For each cluster, one golden summary is created by five NLP-background Ph.D. students or equivalent professionals.

WCEP. Wikipedia Current Events Portal dataset (WCEP) [41] contains human-written summaries of recent news events. Similar articles are also provided by searching similar articles from Common Crawl News dataset[13] to extend the inputs. As a result, large-scale news articles are obtained. Overall, the WCEP dataset has better alignment with the real-world industrial use cases.

Discussion. The main areas covered by the multi-document summarization dataset are news, Wikipedia articles, scientific paper and reviews. In the early development of the multi-document summarization task, the DUC and TAC datasets are the most commonly used. In recent years, large-scale datasets such as Wikisum and Multi-news datasets are used by research works, which indicates the rising trend of data-driven approaches.

7 FUTURE RESEARCH DIRECTIONS AND OPEN ISSUES

It is observed that multi-document summarization techniques are very beneficial in practical applications including news summarization, scientific papers summarization, Wikipedia articles generation, and product reviews summarization. Although the existing works have established a solid foundation for multi-document summarization, it is still a less researched topic compared to single document summarization and other natural language processing topics. Moreover, individuals and industries have a huge demand for compressing multiple documents related to the subject. Thus, this section outlines several prospective research directions and open issues that we believe are critical to the present status of the field.

7.1 Capturing Cross-document Relations for Multi-document Summarization

Currently, one of the problems of many existing deep learning based multi-document summarization models is they center on their simple concatenation of input documents into a flat sequence, which ignores the relations between documents. However, different from single document summarization, documents from the multi-document summarization task may contain the same, complementary or contradictory information with each other [103]. Discovering cross-document relations, which assists models to extract salient information, improves the coherence and reduce redundancy of summaries[71], and provides a desirable solution for this problem. The research on capturing cross-document relations has begun to gain momentum in the past two years. One of the most widely studied topics in cross-document relations is *graphical models*, which can be easily 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 [71, 132, 141, 142]. To this end, a promising direction would be designing a better mechanism to capture cross-document relations, possibly introduce different graph structures into the attention mechanism in deep learning based models.

7.2 Reinforcement Learning for Multi-document Summarization

Reinforcement learning [84, 85, 112] is a cluster of algorithms to deal with sequential decision problems. The reinforcement learning algorithms are based on the intuitive ideas of dynamic programming according to the Bellman Equation. In a sequential decision problem, state transition dynamics of the environment should be provided in advance. Several existing works [89, 100, 140] modeled the document summarization task as a sequential decision problem and adopted reinforcement learning to tackle the task. However, these models are just applied to single document

[13]https://commoncrawl.org/2016/10/news-dataset-available/

summarization problems. Additionally, these summarization methods are currently based on the model-free reinforcement learning algorithms, in which the model is not necessary to be aware of the environment dynamics. Because essentially these model-free-based agents continuously explore the environment through a simple trial-and-error strategy. But they inevitably suffer from low sample efficiency issues. 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 could be potentially explored in the future for multi-document summarization task.

7.3 Pretrained Language Models for Multi-document Summarization

In many natural language processing tasks, the limited labeled corpora, which are not adequate to train semantic-rich word vectors. Utilizing large-scale and unlabeled corpora, which are agnostic with current tasks, for pretraining can enhance the generalization ability of the model and speed up the convergence of the network, as shown in [83, 101]. At present, pretrained language models have led to successes in many deep learning based natural language processing tasks. Among the reviewed papers [65, 71, 156], multiple works adopt pretrained language models for multi-document summarization and achieve promising improvements. Applying pretrained language models such as BERT [28], GPT-2 [107], GPT-3 [15], RoBERTa [75], XLNet [139], ALBERT [64], PEGASUS [147] and fine tuning them on a variety of downstream tasks allows the model achieve faster convergence speed and can improve model performance. Multi-document summarization requires the model to have a strong ability to process long sequences. It is promising to explore powerful language models specifically targeting at the long sequence input characteristic, such as Longformer [12], REFORMER [61], Big bird [144] with pretrained models.

7.4 Explainable Deep Learning Model for Multi-document Summarization

Deep learning models can be regarded as black boxes with high non-linearity. It is extremely challenging to understand the detailed transformation inside deep learning models. However, the interpretability of models is highly crucial for model building. It may target answering how the model generates the candidate summaries. The explainable multi-document summarization model helps distinguish whether the model has learned the distribution of generating condensed and coherent summaries, as well as prevent the bias of the model. It also helps the model designer build models in a better fashion. Recently, a large amount of works [109, 148] of explainable models are proposed to ease the non-interpretable concern of deep neural networks, within which model attention [114, 157] plays an especially important role in model interpretation. Also, explainable methods [53, 63] are intensively researched in the natural language processing area. However, in the multi-document summarization task, the works of explainable models are relatively scarce. Thus, the model explanation can be developed in future works.

7.5 Adversarial Attack and Defense for Multi-document Summarization

Adversarial examples are strategically modified samples that aim to fool the deep neural networks based models. An adversarial example is created via the worst-case perturbation of the input that an ideal or 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 was originated from research in Computer Vision [121] and was introduced in Natural Language Processing by the work [55]. An essential purpose for generating adversarial examples for neural networks is to utilize these adversarial examples to enhance the model's robustness [43]. Therefore, research on adversarial examples is not only beneficial to identify and apply a robust model but also helps to build robust models for different tasks. Following the pioneering work in [55], many attack methods have been proposed to address this problem in various NLP applications [151]. However, very limited research attack DNNs for multi-document summarization [21]. It is worth filling this gap by exploring existing and developing new adversarial attacks on the state-of-the-art DNN-based multi-document summarization models.

7.6 Multi-modality for Multi-document Summarization

Multi-modal learning has led to successes in many deep learning tasks, such as Visual Language Navigation [133] and Visual Question Answering [6]. The works of multi-document summarization combining with multi-modality have a variety of applications: (i) text + image: generating summaries with pictures and texts for those documents with pictures. This kind of multi-modal summary can improve the satisfaction of users [158], (ii) text + video: Based on the video and its subtitles, generating a concise text summary that describes the main context of video [97]. Movie synopsis is one of the applications, and (iii) text + audio: generating short text summary of audio files that people could quickly preview the content without actually listening to the audio recording from the beginning to the end [33]. Deep learning is well suited to the multi-modality task as it is able to effectively capture nonlinear relationships whether on images, text or video data. However, at present, there is little multi-modal research work based on multi-document summarization. This is a promising but largely under-explored area where more studies are expected.

7.7 Improving Evaluation Metrics for Multi-document Summarization

In terms of the evaluation metrics of multi-document summarization task, current evaluation metrics still have several obvious defects. Take the most commonly used metrics ROUGE as an example, despite the fact that there are researches show the effectiveness of the ROUGE-based metrics, they are still challenging to accurately measure the similarity between a generated summary and the golden ones semantically. This is because ROUGE-based evaluation metrics only consider vocabulary-level distances. In other words, even if the ROUGE scores are improved, it does not represent the summary is of a higher quality. It is thus not very ideal for model training. Recently, some works extend ROUGE along with WordNet [115] or pretrained language models [150] tending to alleviate the drawbacks. However, it is challenging to propose evaluation indicators that can reflect the true quality of generated summaries comprehensively and semantically as human raters. Meanwhile, some existing works [39, 120] concentrate on unsupervised evaluation, which is another front line for evaluation metrics research. A good evaluation indicator is able to reflect the true performance of the multi-document summarization model and guide the design of models.

7.8 Creating More Datasets for Multi-document Summarization

Benchmark datasets allow training, evaluating and comparing the capabilities of different models on the same stage. High-quality datasets are critical to the development of the multi-document summarization task. However, in multi-document summarization task, the most commonly used DUC and TAC datasets have a relatively small amount of samples. It is thus not very suitable for training deep neural network models. In recent years, some large datasets have been proposed, including Wikisum [73], Multi-news [35] and WCEP [41], but more efforts are still needed. In the future, datasets with documents of rich diversity are desperately required to promote the research of multi-document summarization. Meanwhile, according to application requirements, datasets in cross-domains are ought to be collected, for example, medical records or dialogue summarization [86], email summarization [124, 145], code summarization [80, 108], software project activities summarization [2], legal documents summarization [57]. The development of large-scale cross-task datasets will facilitate multi-task learning [34, 135]. However, the datasets of multi-document summarization combining with text classification, question answering, or other language tasks have seldom been proposed in the multi-documentation research community, but these datasets are essential and widely employed in industrial applications.

8 CONCLUSION

In this article, we present the first comprehensive review of the most notable works to date on deep learning based multi-document summarization. We propose a taxonomy scheme for organizing and clustering existing publications and devise the network design strategies based on the state-of-the-art methods. Furthermore, we also provide an overview of the existing multi-document objective functions, evaluation metrics and datasets. Additionally, some of the most pressing open problems and promising future extensions are also discussed in this survey. We hope this survey can provide readers with a comprehensive understanding of the key aspects of the multi-document summarization task, clarify the most notable advances, and shed some 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] 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 2020)*. Leiden, The Netherlands.

[3] Reinald Kim Amplayo and Mirella Lapata. 2019. Informative and Controllable Opinion Summarization. *arXiv preprint arXiv:1909.02322*.

[4] 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 2018)*. Brussels, Belgium.

[5] Diego Antognini and Boi Faltings. 2019. Learning to Create Sentence Semantic Relation Graphs for Multi-Document Summarization. *arXiv preprint arXiv:1909.12231*.

[6] 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 2015)*. Santiago, Chile.

[7] Rachit Arora and Balaraman Ravindran. 2008. Latent Dirichlet Allocation and Singular Value Decomposition based Multi-document Summarization. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM 2008)*. Pisa, Italy.

[8] 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 2015)*. San Diego, CA, United States.

[9] 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 (ACL 2005)*.

[10] 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 (SAC2012)*. Riva, Italy.

[11] Phyllis B Baxendale. 1958. Machine-made Index for Technical Literature - An Experiment. *IBM Journal of Research and Development* 2, 4, 354–361.

[12] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The Long-document Transformer. *arXiv preprint arXiv:2004.05150*.

[13] Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2019. Unsupervised Multi-Document Opinion Summarization as Copycat-Review Generation. *arXiv preprint arXiv:1911.02247*.

[14] Arthur Brazinskas, Mirella Lapata, and Ivan Titov. 2019. Unsupervised Multi-Document Opinion Summarization as Copycat-Review Generation. *arXiv preprint arXiv:1911.02247*.

[15] 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. *arXiv preprint arXiv:2005.14165*.

[16] 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 2017)*. San Francisco, United States.

[17] 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 2015)*. Austin, United States.

[18] 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 2015)*. Beijing, China.

[19] 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 1998)*. Melbourne, Australia.

[20] 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 2007)*. Banff, Canada.

[21] 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 2020)*. New York, NY, United States.

[22] 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 2019)*. Florence, Italy.

[23] 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 2013)*. Atlanta, Georgia.

[24] 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 2019)*. Long Beach, United States.

[25] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv preprint arXiv:1412.3555*.

[26] 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*. Hong Kong, China.

[27] 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, Aug, 2493–2537.

[28] 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 2019)*. Minneapolis, United States.

[29] 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 2014)*. Baltimore, United States.

[30] Carl Doersch. 2016. Tutorial on Variational Autoencoders. *arXiv preprint arXiv:1606.05908*.

[31] 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 2014)*. Dublin, Ireland.

[32] 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.

[33] 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 2003)*. Baltimore, United States.

[34] Andre Esteva, Anuprit Kale, Romain Paulus, Kazuma Hashimoto, Wenpeng Yin, Dragomir Radev, and Richard Socher. 2020. Co-search: Covid-19 Information Retrieval with Semantic Search, Question Answering, and Abstractive Summarization. *arXiv preprint arXiv:2006.09595*.

[35] 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 2019)*. Florence, Italy.

[36] 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.

[37] 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 2010)*. Beijing, China.

[38] 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 2019)*. Hong Kong, China.

[39] 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 2020)*. Online.

[40] 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 2014)*. Doha, Qatar.

[41] 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 2020)*. Online.

[42] Jade Goldstein, Vibhu O Mittal, Jaime G Carbonell, and Mark Kantrowitz. 2000. Multi-document Summarization by Sentence Extraction. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Applied Natural Language Processing Conference (NAACL-ANLP 2000)*. Seattle, United States.

[43] 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 2015)*. San Diego, CA, United States,.

[44] 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 2016)*. Berlin, Germany.

[45] 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.

[46] 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 2009)*. Boulder, United States.

[47] M 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.

[48] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the Dimensionality of Data with Neural Networks. *science* 313, 5786, 504–507.

[49] 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 2018)*. Brussels, Belgium.

[50] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-term Memory. *Neural Computation* 9, 8, 1735–1780.

[51] Ya-Han Hu, Yen-Liang Chen, and Hui-Ling Chou. 2017. Opinion Mining from Online Hotel Reviews–A Text Summarization Approach. *Information Processing & Management* 53, 2, 436–449.

[52] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv preprint arXiv:1508.01991*.

[53] 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 2020)*. Online.

[54] Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. 1977. Perplexity - A Measure of the Difficulty of Speech Recognition Tasks. *The Journal of the Acoustical Society of America* 62, S1, S63–S63.

[55] 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 2017)*. Copenhagen, Denmark.

[56] 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 2020)*. Online.

[57] Ambedkar Kanapala, Sukomal Pal, and Rajendra Pamula. 2019. Text Summarization from Legal Documents: A Survey. *Artificial Intelligence Review* 51, 3, 371–402.

[58] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*. Doha, Qatar.

[59] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR 2014)*. Banff, Canada.

[60] 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 2017)*. Toulon, France.

[61] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The Efficient Transformer. In *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020)*. Addis Ababa, Ethiopia.

[62] 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 2012)*. Lake Tahoe, United States.

[63] 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 2020)*. Online.

[64] 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 2020)*. Addis Ababa, Ethiopia.

[65] 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 2019)*. Florence, Italy.

[66] 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 2018)*. Brussels, Belgium.

[67] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based Learning Applied to Document Recognition. *Proc. IEEE* 86, 11, 2278–2324.

[68] 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@EMNLP 2017)*. Copenhagen, Denmark.

[69] 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 2017). Copenhagen, Denmark.

[70] Piji Li, Zihao Wang, Wai Lam, Zhaochun Ren, and Lidong Bing. 2017. Salience Estimation via Variational Auto-encoders for Multi-document Summarization. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI 2017)*. San Francisco,United States.

[71] 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 2020)*. Online.

[72] Chin-Yew Lin. 2004. Rouge: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. 74–81.

[73] 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 2018)*. Vancouver,Canada.

[74] 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 2019)*. Florence, Italy.

[75] 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*.

[76] Annie Louis and Ani Nenkova. 2013. Automatically Assessing Machine Summary Content Without A Gold Standard. *Computational Linguistics* 39, 2, 267–300.

[77] Wencan Luo and Diane Litman. 2015. Summarizing Student Responses to Reflection Prompts. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*. Lisbon, Portugal.

[78] 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 2016)*. San Diego California, United States.

[79] Inderjeet Mani and Eric Bloedorn. 1997. Multi-Document Summarization by Graph Search and Matching. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Innovative Applications of Artificial Intelligence Conference (AAAI 1997)*. Providence, United States.

[80] 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 2014)*. Hyderabad, India.

[81] 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 2016)*. Austin, United States.

[82] 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 2005)*. Jeju Island, Republic of Korea.

[83] 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 2013)*. Lake Tahoe, United States, 3111–3119.

[84] 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 (ICML2016)*. New York City, United States.

[85] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. *arXiv preprint arXiv:1312.5602*.

[86] 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 2020)*. Grenoble, France.

[87] N Moratanch and S Chitrakala. 2016. A Survey on Abstractive Text Summarization. In *Proceedings of the 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT 2016)*. Nagercoil, India.

[88] 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 Thirty-First AAAI Conference on Artificial Intelligence (AAAI 2017)*. San Francisco, United States.

[89] 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 2018)*. New Orleans, United States.

[90] 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 2018)*. Santa Fe, United States.

[91] 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 2018)*. Santa Fe, United States.

[92] Ani Nenkova and Kathleen R. McKeown. 2012. A Survey of Text Summarization Techniques. In *Mining Text Data*.

[93] 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–es.

[94] Ani Nenkova and Rebecca J Passonneau. 2004. Evaluating Content Selection in Summarization: The Pyramid Method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2004)*. Boston, United States.

[95] Jun-Ping Ng and Viktoria Abrecht. [n.d.]. Better Summarization Evaluation with Word Embeddings for ROUGE. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*.

[96] 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.

[97] 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 2019)*. Florence, Italy.

[98] 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 (ACL2002)*. Philadelphia, United States.

[99] 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 2013)*. Sofia, Bulgaria.

[100] 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 2018)*. Vancouver, Canada.

[101] 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 2018)*. New Orleans, United States.

[102] Maxime Peyrard. 2019. A Simple Theoretical Model of Importance for Summarization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL 2019)*. Association for Computational Linguistics, Florence, Italy, 1059–1073.

[103] 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 2000)*. Hong Kong, China.

[104] Dragomir R Radev, Eduard Hovy, and Kathleen McKeown. 2002. Introduction to the Special Issue on Summarization. *Computational linguistics* 28, 4, 399–408.

[105] Dragomir R Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. 2004. Centroid-based Summarization of Multiple Documents. *Information Processing & Management* 40, 6, 919–938.

[106] Dragomir R. Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara. 2013. The ACL Anthology Network Corpus. *Lang. Resour. Evaluation* 47, 4, 919–944.

[107] 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.

[108] 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 2014)*. Hyderabad, India.

[109] 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.

[110] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning Representations by Back-propagating Errors. *Nature* 323, 6088, 533–536.

[111] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic Routing Between Capsules. In *Proceedings of the 2017 Annual Conference on Neural Information Processing Systems(NIPS 2017)*. Long Beach, United States, 3856–3866.

[112] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*.

[113] 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 2017)*. Vancouver, Canada.

[114] Sofia Serrano and Noah A. Smith. 2019. Is Attention Interpretable?. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL 2019)*. Florence, Italy, 2931–2951.

[115] Elaheh ShafieiBavani, Mohammad Ebrahimi, Raymond 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 2018)*. 762–767.

[116] 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*. 442–451.

[117] Ori Shapira, David Gabay, Yang Gao, Hadar Ronen, Ramakanth Pasunuru, Mohit Bansal, Yael Amsterdamer, and Ido Dagan. [n.d.]. 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 2019)*.

[118] Nikhil S Shirwandkar and Samidha Kulkarni. 2018. Extractive Text Summarization Using Deep Learning. In *Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA 2018)*. Pune, India.

[119] 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 2018)*. New Orleans, United States.

[120] 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 2019)*. Hong Kong, China.

[121] 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 2014)*. Banff, Canada.

[122] Oguzhan Tas and Farzad Kiyani. 2007. A Survey Automatic Text Summarization. *PressAcademia Procedia* 5, 1, 205–213.

[123] 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 2016)*. Berlin, Germany.

[124] Jan Ulrich, Gabriel Murray, and Giuseppe Carenini. 2008. A Publicly Available Annotated Corpus for Supervised Email Summarization. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence in Enhanced Messaging Workshop (AAAI 2008)*. Chicago, United States.

[125] 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 Annual Conference on Neural Information Processing Systems (NIPS 2017)*. Long Beach, United States.

[126] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph Attention Networks. *arXiv preprint arXiv:1710.10903*.

[127] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer Networks. In *Proceedings of the 2015 Conference of Advances in Neural Information Processing Systems (NIPS 2015)*. Montreal, Canada.

[128] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer Networks. In *Proceedings of the 2015 Annual Conference on Neural Information Processing Systems (NIPS 2015)*. Montreal, Canada.

[129] 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 2013)*. Salford, UK.

[130] Xiaojun Wan and Jianwu Yang. 2006. Improved Affinity Graph based Multi-document Summarization. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (NAACL 2006)*. New York,United States.

[131] 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 2008)*. Singapore.

[132] Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020. Heterogeneous Graph Neural Networks for Extractive Document Summarization. *arXiv preprint arXiv:2004.12393*.

[133] 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 2020)*. Online.

[134] 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 2016)*. San Diego California, United States.

[135] 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 2020)*. Online, 3586–3596.

[136] 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 2020)*. Online.

[137] 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 2018)*. Brussels, Belgium.

[138] Qian Yang, Rebecca J Passonneau, and Gerard De Melo. 2016. PEAK: Pyramid Evaluation via Automated Knowledge Extraction. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI 2016)*. Phoenix, Arizona.

[139] 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 Annual Conference on Neural Information Processing System (NIPS 2019)*. Vancouver, Canada.

[140] Kaichun Yao, Libo Zhang, Tiejian Luo, and Yanjun Wu. 2018. Deep Reinforcement Learning for Extractive Document Summarization. *Neurocomputing* 284, 52–62.

[141] 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 AAAI Conference on Artificial Intelligence (AAAI 2019)*. Honolulu, United States.

[142] 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 2017)*.

[143] 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 2015)*. Buenos Aires, Argentina.

[144] Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big Bird: Transformers for Longer Sequences. *arXiv preprint arXiv:2007.14062*.

[145] 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 & Management* 44, 4, 1600–1610.

[146] 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 2018)*. Tilburg, Netherlands.

[147] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J Liu. 2019. Pegasus: Pre-training with Extracted Gap-sentences for Abstractive Summarization. *arXiv preprint arXiv:1912.08777*.

[148] 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 (CVPR2018)*. Salt Lake City, United States.

[149] 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 2020)*. Addis Ababa, Ethiopia.

[150] 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 2020)*. Addis Ababa, Ethiopia.

[151] 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.

[152] 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.

[153] Guangxiang Zhao, Junyang Lin, Zhiyuan Zhang, Xuancheng Ren, Qi Su, and Xu Sun. 2019. Explicit Sparse Transformer: Concentrated Attention Through Explicit Selection. *arXiv preprint arXiv:1912.11637*.

[154] 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 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*. Hong Kong, China.

[155] 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 2019)*. Hong Kong, China.

[156] 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 2020)*. Online.

[157] 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 2016)*. Las Vegas, United States.

[158] 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 2018)*. Brussels, Belgium.

[159] 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 2018)*. New Orleans, United States.