THE UNIVERSITY of ADELAIDE

Deep Learning Based Multi-document Summarization

Congbo Ma

A thesis submitted for the degree of
DOCTOR OF PHILOSOPHY
The University of Adelaide

February 6, 2024

Contents

List of Figures

List of Tables

Publications

This thesis contains the following works that have been published or prepared for publication:

- Multi-document Summarization via Deep Learning Techniques: A Survey.
  Congbo Ma, Wei Emma Zhang, Mingyu Guo, Hu Wang, Quan Z Sheng.
  ACM Computing Surveys (CSUR), 2022.

- Incorporating Linguistic Knowledge for Abstractive Multi-document Summarization.
  Congbo Ma, Wei Emma Zhang, Hu Wang, Shubham Gupta, Mingyu Guo.
  Proceedings of the 36th Pacific Asia Conference on Language, Information and Computation (PACLIC), 2022.

- Document-aware Positional Encoding and Linguistic-guided Encoding for Abstractive Multi-document Summarization.
  Congbo Ma, Wei Emma Zhang, Pitawelayalage Dasun Dileepa Pitawela, Yutong Qu, Haojie Zhuang, Hu Wang.
  IEEE World Congress on Computational Intelligence - International Joint Conference on Neural Networks (WCCI-IJCNN), 2022.

- Improving deep learning based multi-document summarization through linguistic knowledge.
  Congbo Ma.
  Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2021.

- Exploring Transformer-based Multi-document Summarization: An Empirical Investigation.
  Congbo Ma, Wei Emma Zhang, Pitawelayalage Dasun Dileepa Pitawela, Haojie Zhuang, Yanfeng Shu.
  In Submission.

- Disentangling Specificity for Abstractive Multi-document Summarization.
  Congbo Ma, Wei Emma Zhang, Hu Wang, Haojie Zhuang, Mingyu Guo.
  In Submission.

In addition, during my Ph.D., I have the following papers not included in this thesis:

- Learnable Cross-modal Knowledge Distillation for Multi-modal Learning with Missing Modality.
  Hu Wang, **Congbo Ma**, Jianpeng Zhang, Yuan Zhang, Jodie Avery, Louise Hull, Gustavo Carneiro.
  International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2023.

- Multi-Modal Learning With Missing Modality via Shared-Specific Feature Modelling.
  Hu Wang, Yuanhong Chen, **Congbo Ma**, Jodie Avery, Louise Hull, Gustavo Carneiro.
  Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023.

- Learning From the Source Document: Unsupervised Abstractive Summarization.
  Haojie Zhuang, Wei Emma Zhang, Jian Yang, **Congbo Ma**, Yutong Qu, Quan Z Sheng.
  Findings of Conference on Empirical Methods in Natural Language Processing (EMNLP), 2022.

- An Empirical Study on Topic Preservation in Multi-Document Summarization.
  Mong Yuan Sim, Wei Emma Zhang, **Congbo Ma**.
  Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: Student Research Workshop (AACL/IJCNLP), 2022.

- Uncertainty-aware Multi-modal Learning via Cross-modal Random Network Prediction.
  Hu Wang, Jianpeng Zhang, Yuanhong Chen, **Congbo Ma**, Jodie Avery, Louise Hull, Gustavo Carneiro.
  The European Conference on Computer Vision (ECCV), 2022.

- The 10 research topics in the Internet of Things.
  Wei Emma Zhang, Quan Z Sheng, Adnan Mahmood, Munazza Zaib, Salma Abdalla Hamad, Abdulwahab Aljubairy, Ahoud Abdulrahmn F Alhazmi, Subhash Sagar, **Congbo Ma**.
  IEEE 6th International Conference on Collaboration and Internet Computing (CIC), 2020.

University of Adelaide

Abstract

Deep Learning Based Multi-document Summarization

by Congbo Ma

In this era of rapidly advancing technology, the exponential increase of data availability makes analyzing and understanding text files a tedious, labor-intensive, and time-consuming task. Multi-document summarization (MDS) is an effective tool for information aggregation that generates an informative and concise summary from a cluster of topic-related documents. In this thesis, we systematically over-viewed the recent deep learning based MDS models, proposed a series of novel methods to cope with the MDS tasks with deep learning technique and examine the behaviours of Transformer-based MDS models.

Firstly, we presented a categorization scheme to organize current research and provide a comprehensive review for deep learning based MDS techniques, including deep learning based models, objective functions, benchmark datasets, and evaluation metrics. We reviewed development movements and provide a systematic overview and summary of the state-of-the-art. We also summarized nine network design strategies based on our extensive studies of the current models.

Secondly, due to linguistic knowledge plays an important role in assisting models to learn informative representations, in this thesis, we presented a Transformer-based abstractive MDS method with linguistic-guided attention (LGA) mechanism for better representation learning. The proposed linguistic-guided attention mechanism can be seamlessly incorporated into multiple mainstream Transformer based summarization models to improve the quality of the generated summaries. We developed the proposed method based on Flat Transformer (FT) and Hierarchical Transformer (HT), named ParsingSum-FT and ParsingSum-HT respectively. Based on this work, we further proposed document-aware positional encoding and linguistic-guided encoding that can be fused with Transformer architecture for MDS. For document-aware positional encoding, we introduced a general protocol to guide the selection of document encoding functions. For linguistic-guided encoding, we presented to embed syntactic dependency relations into the dependency relation mask with a simple

but effective non-linear encoding learner for feature learning. Empirical studies on both models demonstrate these two simple but effective methods can help the models outperform existing Transformer-based methods on the benchmark dataset by a large margin.

Thirdly, the existing MDS methods neglect the specific information for each document, limiting the comprehensiveness of the generated summaries. To solve this problem, we presented to disentangle the specific content from documents in one document set. The document-specific representations, which are encouraged to be distant from each other via a proposed orthogonal constraint, are learned by the specific representation learner. We provided extensive analysis and had interesting findings that specific information can well-complementary with document set features for MDS tasks. Also, we found that the common (i.e. shared) information could not contribute much to the overall performance under the MDS settings.

Fourthly, the utilization of Transformer based models prospers the growth of MDS. In order to thoroughly examine the behaviours of Transformer based MDS models, this thesis also presented five empirical studies on (1) measuring the impact of document separators quantitatively; (2) exploring the effectiveness of different mainstream Transformer structures; (3) examining the sensitivity of encoder and decoder (4) discussing different training strategies; (5) discovering the repetition in summary generation. The experimental results on two MDS datasets and eleven evaluation metrics show the influence of document separators, the granularity of different level features and different model training strategies. The experiments also indicated that the decoder exhibits greater sensitivity to noises in summarization tasks compared to the encoder, which indicates the important role played by the decoder, pointing a potential direction for future MDS researches. Furthermore, the experimental results indicated that the repetition problem in the generated summaries have correlations with the high uncertainty score.

Finally, we discussed the open issues of deep learning based MDS and identified the future research directions of this field. We also proposed potential solutions for some discussed research directions.

Declaration of Authorship

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint award of this degree. The author acknowledges that copyright of published works contained within the thesis resides with the copyright holder(s) of those works. I give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time. I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Congbo Ma

Oct 2023

Acknowledgements

First and foremost, I would like to express my heartfelt gratitude to my Ph.D. supervisor, Dr. Wei Zhang, for her unwavering support and guidance throughout my doctoral journey. Her unwavering commitment to nurturing my growth as a researcher has played a pivotal role in shaping my academic pursuits. The countless discussions, paper revisions, and collaborative endeavors are all experiences that I will forever hold dear. Dr. Wei Zhang is not only a dedicated mentor but also a friend on this intellectual voyage. She not only gave me a lot of guidance in scientific research but also helped me a lot in life. I feel profoundly honored and deeply grateful to have had the privilege of being mentored by such an outstanding individual.

I'd like to express my sincere gratitude to Dr. Mingyu Guo and Dr. Weitong Chen, my co-supervisors during my Ph.D. journey. Their patient support has been a great source of encouragement that has motivated me to persevere in my research endeavors.

I would like to thank my closely worked collaborators: Prof. Michael Sheng, Dr. Hu Wang, Mr Haojie Zhuang, Miss Yutong Qu, Mr Pitawelayalage Dasun Dileepa Pitawela, Mr Shubham Gupta. I appreciate their assistance and every insightful discussion we had. These discussions have truly illuminated my research journey.

Many thanks go to all my supportive friends. Our time here in Adelaide has been nothing short of wonderful, and I've cherished every moment spent with them. I would also like to express my gratitude to Ms Jane Garrard and Mr Rick Garrard, whom we met through talking with Aueesie event. Their introduction to Australian life and thoughtful care made me feel like I was with family, and I'm really grateful for that.

I would like to express my heartfelt gratitude to my dear parents. Their constant encouragement, patience, and unwavering understanding are the source of my happiness. Their support is the driving force behind my achievements, and I consider myself incredibly fortunate to have you by my side. I owe a deep sense of gratitude to my husband. His belief in me and his endless support have made all the difference. I am profoundly grateful to have him in my life.

Last but not least, I would like to extend my heartfelt gratitude to the University of Adelaide for providing me with the exceptional academic environment, resources, and opportunities that made this doctoral journey possible. The dedication and support of the faculty, staff, and students have enriched my learning experience and contributed significantly to my research.

Chapter 1

Introduction

1.1 Background and Motivations

A short and concise summary can be generated from one or several lengthy documents, resulting in single document summarization (SDS) and multi-document summarization (MDS). While simpler to perform, SDS may not produce comprehensive summaries because it does not take several related, or more recent, documents into account. Conversely, multi-document summarization generates more comprehensive and accurate summaries from documents written at different times, covering different perspectives, but is accordingly more complicated as it tries to resolve potentially diverse and redundant information (Tas and Kiyani, 2007). Formally, 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_{d^i} sentences $\{s_{i,j} \mid j \in [1, M_{d^i}]\}$. $s_{i,j}$ refers to the j-th sentence in the i-th document. The standard summary Ref is called the *gold summary* or *reference summary*.

Multi-document summarization enjoys a wide range of real-world applications, including summarization of news (Fabbri et al., 2019a), scientific publications (Yasunaga et al., 2019), emails (Carenini, Ng, and Zhou, 2007; Zajic, Dorr, and Lin, 2008), product reviews (Gerani et al., 2014), medical documents (Afantenos, Karkaletsis, and Stamatopoulos, 2005; Wang et al., 2023), lecture feedback (Luo et al., 2016), software project activities (Alghamdi, Treude, and Wagner, 2020), and Wikipedia articles (Liu et al., 2018a). Multi-document summarization technology has also received a great amount of industry attention; an intelligent multilingual news reporter bot named Xiaomingbot (Xu et al., 2020b) was developed for news generation, which can summarize multiple news sources into one article and translate it into multiple languages. Massive application requirements and rapidly growing online data have promoted the development of multi-document summarization.

Existing traditional MDS algorithms are based on: term frequency-inverse document frequency (TF-IDF) (Radev et al., 2004; Baralis et al., 2012), clustering (Goldstein et al., 2000; Wan and Yang, 2008), graphs (Mani and Bloedorn, 1997; Wan and Yang, 2006) and latent semantic analysis (Arora and Ravindran, 2008; Haghighi and Vanderwende, 2009). Most of these works still generate summaries with manually crafted features (Mihalcea and Tarau, 2005; Wan and Yang, 2006), such as sentence position features (Baxendale, 1958; Erkan and Radev, 2004a), sentence length features (Erkan and Radev, 2004a), proper noun features (Vodolazova et al., 2013), cue-phrase features (Gupta and Lehal, 2010), biased word features, sentence-to-sentence cohesion and sentence-to-centroid cohesion.

More recently, deep learning has gained enormous attention due to its success in various domains, for instance, computer vision (Krizhevsky, Sutskever, and Hinton, 2012), natural language processing (Devlin et al., 2014) and multi-modal learning (Im et al., 2021). Both industry and academia have embraced deep learning to solve complex tasks due to its capability of mapping highly nonlinear relations from data to the labels. Deep neural network models learn multiple levels of representation and abstraction from input data. Deep learning algorithms replace manual feature engineering by capturing distinctive features through back-propagation to minimize given objective functions. It is well known that linear solvable problems possess many advantages, such as it can be easily solved and has numerous theoretically proven supports; However, many NLP tasks are highly non-linear. As theoretically proven by Hornik et al. (Hornik, Stinchcombe, and White, 1989), neural networks can fit any given continuous function as a universal approximator. For multi-document summarization tasks, deep neural networks also perform considerably better than traditional methods to effectively process large-scale documents and distill informative summaries due to their strong fitting abilities. Therefore, deep learning based methods demonstrate outstanding performance in multi-document summarization tasks in most cases (Cao et al., 2015b; Liu and Lapata, 2019a; Lebanoff et al., 2019; Lu, Dong, and Charlin, 2020a; Li et al., 2020b; Chen et al., 2021a; Xiao et al., 2022; Wen et al., 2022; Moro et al., 2022; Puduppully et al., 2023; Atri et al., 2023; Amar et al., 2023). With recent huge improvements in computational power and the release of increasing numbers of public datasets, neural networks with deeper layers and more complex structures have been applied in multi-document summarization (Liu and Lapata, 2019a; Li et al., 2017b), accelerating the development of text summarization with more powerful and robust models. The prosperity of deep learning for summarization in both academia and industry requires a comprehensive review of current publications for researchers to better understand the process and research progress. However, most of the existing summarization survey papers are based on traditional

algorithms instead of deep learning based methods or target general text summarization (Nenkova and McKeown, 2012; Haque, Pervin, Begum, et al., 2013; Ferreira et al., 2014; Shah and Jivani, 2016; El-Kassas et al., 2021). We have therefore investigated recent publications on deep learning based MDS methods in this thesis. We classified neural based MDS techniques into diverse categories thoroughly and systematically, and we also conducted a detailed discussion on the categorization and progress of these approaches to establish a clearer concept standing in the shoes of readers. Based on the existing MDS works, we found some challenges that deep learning based multi-document summarization models faced:

- Deep learning methods often lack crucial linguistic knowledge, limiting their ability to assist learners in creating informative representations and guiding summary generation effectively. We believe that this is one possible reason that some non-deep learning based MDS methods sometimes show better performance than deep learning based methods (Lu, Dong, and Charlin, 2020a; Cao et al., 2015b) as non-deep learning based methods pay more attention to linguistic information.

- MDS faces a challenge related to the handling of document-specific details. In a collection of documents, each document in a set describes topic-relevant concepts, while per document also has its unique contents. Unfortunately, existing MDS approaches tend to overlook the document-specific aspects, leading to a lack of comprehensiveness in the generated summaries.

- Deep learning based models can be regarded as black boxes with high non-linearity. It is challenging to understand the detailed transformation inside. The contemporary developments of Transformer architecture (Vaswani et al., 2017) thrives MDS task. Exploring the behaviours of Transformer-based MDS models allows researchers to understand the effects of each module in these models, therefore guiding the model design with a more accurate target.

Facing aforementioned challenges and moving towards the solutions, this thesis systematically proposed corresponding methods and analysis of multi-document summarization. We first presented a simple yet effective linguistic-guided attention mechanism for integrating dependency relations within multi-head attention mechanisms. This linguistic-guided attention mechanism can be seamlessly integrated into various mainstream Transformer-based summarization models, resulting in substantial performance enhancements. Based on this work, we extended our efforts by encoding 45 distinct dependency relations into a dependency relation mask using a

straightforward yet highly effective non-linear encoding strategy aimed at enhancing feature learning. Additionally, we incorporated document positional information to assist models in capturing cross-document relations. We conducted an extensive analysis encompassing various configurations of document-aware positional encoding and linguistic-guided encoding.

In order to address the second challenge, our intuition is not only to capture the overall information in a document set but also to distinguish the specificity of each document and learn representations of document specificity which will be considered in the summary generation process. To this end, we presented disentangling specificity for abstractive multi-document summarization (DisentangleSum) — a simple yet effective summarization model that disentangles document uniqueness with a set of document-specific representation learners. In order to optimize the learning of specific features, we further proposed an orthogonal constraint to encourage the specific features obtained from a pair of documents to be distinctive from each other. This constraint encourages the document-specific feature vectors to align vertically with each other, ensuring a semantic separation between them. Based on the constraint, we designed an objective function that can transform the exponential increment of the loss computation between each paired of documents into linear to cope with a large number of documents in a set. Experimental results on two MDS datasets demonstrate the effectiveness of DisentangleSum. We additionally offered comprehensive analyses from multiple perspectives to investigate the underlying mechanisms of DisentangleSum and circumstances of the proposed model that can work.

To solve the third challenge, we undertook a comprehensive investigation from five distinct perspectives covering the Transformer-based multi-document summarization model designing pipeline. (1) Document input perspective: we conducted experiments to quantitatively assess the impact of document separators from the standpoint of document input. (2) Transformer structure perspective: we explored the effectiveness of different mainstream Transformer structures; (3) The significance of encoder and decoder in MDS model: we designed empirical studies by adding noises on top of the encoder and decoder; (4) Training strategy angle: we reorganized the source documents and include self-supervised training techniques; (5) Summary generation angle, we explored the uncertainty when repetition problems occur in summary generation process.

1.2 Thesis Organization

The structure of this thesis is organized as follows: Chapter 1 provides foundational knowledge in the field of multi-document summarization. This chapter encompasses

essential components such as problem definition, applications, typologies of document summarization, research challenges and a brief description of our solutions. In Chapter 2, we structurally overviewed the recent deep learning based multi-document summarization models via a proposed taxonomy. Particularly, we presented a novel mechanism to summarize the design strategies of neural networks and conducted a comprehensive summary. We also highlighted the various objective functions, evaluation metrics and datasets within MDS tasks. Chapter 3 introduced two innovative methods for integrating linguistic knowledge into abstractive multi-document summarization. Initially, a linguistically guided attention mechanism is proposed to incorporate dependency relations within multi-head attention mechanisms, offering a simple yet effective approach. Building upon this foundation, we expanded our endeavors by encoding 45 distinct dependency relations into a dependency relation mask, employing a straightforward yet well-performing non-linear encoding strategy to enrich MDS feature learning. In Chapter 4, we presented DisentangleSum, an innovative MDS model which is capable of disentangling specific information from each document in a set, thereby enhancing the quality of summary generation. Notably, this work represented the first attempt to consider document-specific information in the context of multi-document summarization. Chapter 5 conducted a comprehensive exploration from five distinct angles, encompassing the pipeline for designing Transformer-based MDS models. These perspectives include the document input perspective, Transformer structure perspective, the significance of the encoder and decoder, training strategy angle and summary generation angle. Chapter 6 discussed the future research directions and open issues. Finally, Chapter 7 provided a succinct summary of the thesis.

Chapter 2

Literature Review

This chapter covers various aspects of the advanced deep learning based works in multi-document summarization. Similarities and differences between single document summarization and multi-document summarization are introduced in section 2.1. Nine deep learning architecture design strategies, six deep learning based methods, and the variant tasks of multi-document summarization are presented in section 2.2. Section 2.3 summarized objective functions that guide the model optimization process while evaluation metrics in section 2.4 summarized suitable indices to evaluate the effectiveness of a model. Section 2.5 summarized standard and the variant multi-document summarization datasets.

2.1 From Single to Multi-document Summarization

To have a clear understanding of the processing of deep learning based summarization tasks, we summarized and illustrated the processing framework as shown in Figure 2.1. The first step is preprocessing input document(s), such as segmenting sentences, tokenizing non-alphabetic characters, and removing punctuation (Shirwandkar and Kulkarni, 2018). Multi-document summarization models in particular need to select suitable concatenation methods to capture cross-document relations. Then, an appropriate deep learning based model is chosen to generate semantic-rich

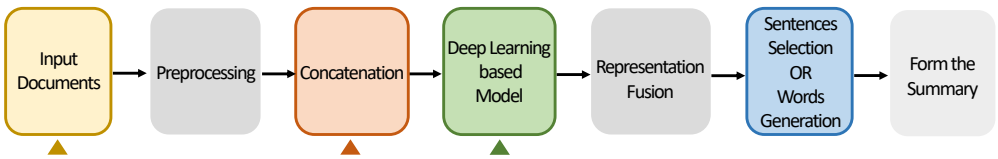


FIGURE 2.1. The processing framework of text summarization. Each of the highlighted steps (the one with the triangle mark) indicates the differences between single document summarization and multi-document summarization.

representation for downstream tasks. The next step is to fuse these various types of representation for later sentence selection or summary generation. Finally, document(s) are transformed into a concise and informative summary. Each of the highlighted steps in Figure 2.1 (indicated by triangles) indicates a difference between single document summarization and multi-document summarization. Based on this process, the research questions of multi-document summarization can be summarized as follows:

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

The following sections provide a comprehensive analysis of the similarities and differences between single document summarization and multi-document summarization.

2.1.1 Similarities between SDS and MDS

Existing single document summarization and multi-document summarization methods share the summarization construction types, learning strategies, evaluation indexes and objective functions. Single document summarization and multi-document summarization both seek to compress the document(s) into a short and informative summary. Existing summarization methods can be grouped into *abstractive summarization*, *extractive summarization* and *hybrid summarization* (Figure 2.2). Extractive summarization methods select salient snippets from the source documents to create informative summaries, and generally contain two major components: *sentence ranking* and *sentence selection* (Cao et al., 2015a; Nallapati, Zhai, and Zhou, 2017). Abstractive summarization methods aim to present the main information of input documents by automatically generating summaries that are both succinct and coherent; this cluster of methods allows models to generate new words and sentences from a corpus pool (Paulus, Xiong, and Socher, 2018). Hybrid models are proposed to combine the advantages of both extractive and abstractive methods to process the

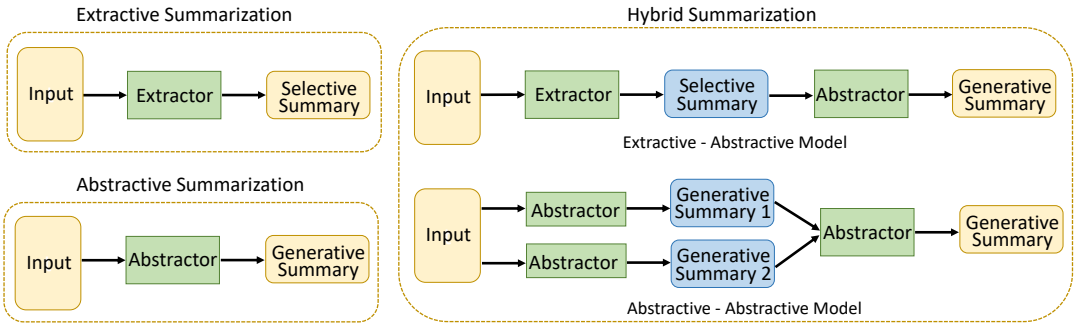


FIGURE 2.2. Summarization construction types for text summarization.

input texts. Research on summarization focuses on two learning strategies. One strategy seeks to enhance the generalization performance by improving the architecture design of the end-to-end models (Fabbri et al., 2019b; Chu and Liu, 2019; Jin, Wang, and Wan, 2020a; Liu and Lapata, 2019a). The other leverages external knowledge or other auxiliary tasks to complement summary selection or generation (Cao et al., 2017; Li et al., 2020b). Furthermore, both single document summarization and multi-document summarization aim to minimize the distance between machine-generated summary and gold summary. Therefore, single document summarization and multi-document summarization could share some indices to evaluate the performance of summarization models such as Recall-Oriented Understudy for Gisting Evaluation (ROUGE), and objective functions to guide model optimization.

2.1.2 Differences between SDS and MDS

In the early stages of multi-document summarization, researchers directly applied single document summarization models to multi-document summarization (Mao et al., 2020). However, a number of aspects in multi-document summarization that are different from single document summarization and these differences are also the breakthrough point for exploring the multi-document summarization models. We summarize the differences in the following five aspects:

- More diverse input document types;
- Insufficient methods to capture cross-document relations;
- High redundancy and contradiction across input documents;
- Larger searching space but lack of sufficient training data;

- Lack of evaluation metrics specifically designed for multi-document summarization.

A defining different character between single document summarization and multi-document summarization is the number of input documents. multi-document summarization tasks deal with multiple sources, of types that can be roughly divided into three groups:

- Many short sources, where each document is relatively short but the quantity of the input data is large. A typical example is product reviews summarization that aims to generate a short, informative summary from numerous individual reviews (Angelidis and Lapata, 2018).

- Few long sources. For example, generating a summary from a group of news articles (Fabbri et al., 2019b), or constructing a Wikipedia-style article from several web articles (Liu et al., 2018a).

- Hybrid sources containing one or few long documents with several to many shorter documents. For example, news article(s) with several readers' comments to this news (Li, Bing, and Lam, 2017), or a scientific summary from a long paper with several short corresponding citations (Yasunaga et al., 2019).

As single document summarization only uses one input document, no additional processing is required to assess relationships between single document summarization inputs. By their very nature, the multiple input documents used in multi-document summarization are likely to contain more contradictory, redundant, and complementary information (Radev, 2000). Multi-document summarization models therefore require sophisticated algorithms to identify and cope with redundancy and contradictions across documents to ensure that the final summary is comprehensive. Detecting these relations across documents can bring benefits for multi-document summarization models. In multi-document summarization tasks, there are two common methods to concatenate multiple input documents:

- Flat concatenation is a simple yet powerful concatenation method, where all input documents are spanned and processed as a flat sequence; to a certain extent, this method converts multi-document summarization to single document summarization tasks. Inputting flat-concatenated documents requires models to have a strong ability to process long sequences.

- Hierarchical concatenation is able to preserve cross-document relations. However, many existing deep learning methods do not make full use of this hierarchical relationship (Wang et al., 2020a; Fabbri et al., 2019b; Liu et al.,

2018a). Taking advantage of hierarchical relations among documents instead of simply flat concatenating articles facilitates the multi-document summarization model to obtain representation with built-in hierarchical information, which in turn improves the effectiveness of the models. The input documents within a cluster describe a similar topic logically and semantically. Figure 2.3 illustrates two representative methods of hierarchical concatenation. Existing hierarchical concatenation methods either perform document-level condensing in a cluster separately (Amplayo and Lapata, 2021) or process documents in word/sentence-level inside document cluster (Nayeem, Fuad, and Chali, 2018; Antognini and Faltings, 2019; Wang et al., 2020a). In Figure 2.3(a), the extractive or abstractive summaries, or representation from the input documents are fused in the subsequent processes for final summaries generation. The models using document-level concatenation methods are usually two-stage models. In Figure 2.3(b), sentences in the documents can be replaced by words. For word or sentence-level concatenation methods, clustering algorithms and graph-based techniques are the most commonly used methods. Clustering methods could help multi-document summarization models decrease redundancy and increase the information coverage for the generated summaries (Nayeem, Fuad, and Chali, 2018). Sentence relation graph is able to model hierarchical relations among multi-documents as well (Antognini and Faltings, 2019; Yasunaga et al., 2019; Yasunaga et al., 2017). Most of the graph construction methods utilize sentences as vertexes and the edge between two sentences indicates their sentence-level relations (Antognini and Faltings, 2019). Cosine similarity graph (Erkan and Radev, 2004a), discourse graph (Christensen, Soderland, Etzioni, et al., 2013; Yasunaga et al., 2017; Liu and Lapata, 2019a), semantic graph (Pasunuru et al., 2021b) and heterogeneous graph (Wang et al., 2020a) can be used for building sentence graph structures. These graph structures could all serve as an external knowledge to improve the performance of multi-document summarization models.

In addition to capture cross-document relation, hybrid summarization models can also be used to capture complex documents semantically, as well as to fuse disparate features that are more commonly adopted by multi-document summarization tasks. These models usually process data in two stages: extractive-abstractive and abstractive-abstractive (the right part of Figure 2.2). The two-stage models try to gather important information from source documents with extractive or abstractive methods at the first stage, to significantly reduce the length of documents. In the second stage, the processed texts are fed into an abstractive model to form final summaries (Amplayo and Lapata, 2021; Lebanoff et al., 2019; Liu et al., 2018a; Liu and

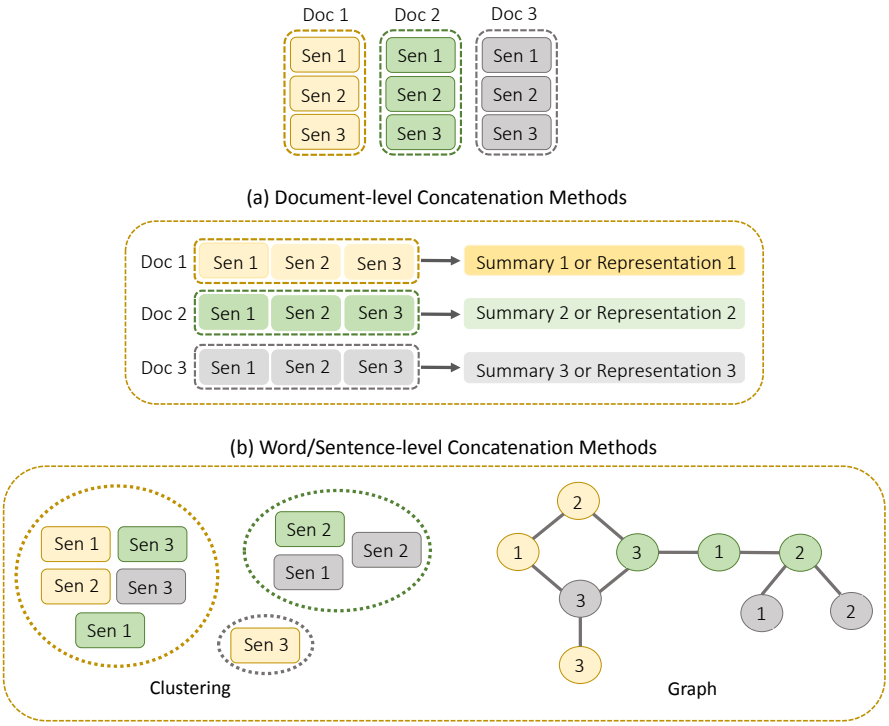


FIGURE 2.3. The methods of hierarchical concatenation.

Lapata, 2019a; Li et al., 2020b).

2.2 Deep Learning Based Multi-document Summarization Methods

Deep neural network (DNN) models learn multiple levels of representation and abstraction from input data and can fit data in a variety of research fields. Deep learning algorithms replace manual feature engineering by learning distinctive features through back-propagation to minimize a given objective function. It is well known that linear solvable problems possess many advantages, such as being easily solved and having numerous theoretically proven supports; however, many NLP tasks are highly non-linear. As theoretically proven by Hornik et al. (Hornik, Stinchcombe, and White, 1989), neural networks can fit any given continuous function as a universal approximator. For multi-document summarization tasks, DNNs also perform considerably better than traditional methods to effectively process large-scale documents and distill informative summaries due to their strong fitting abilities.

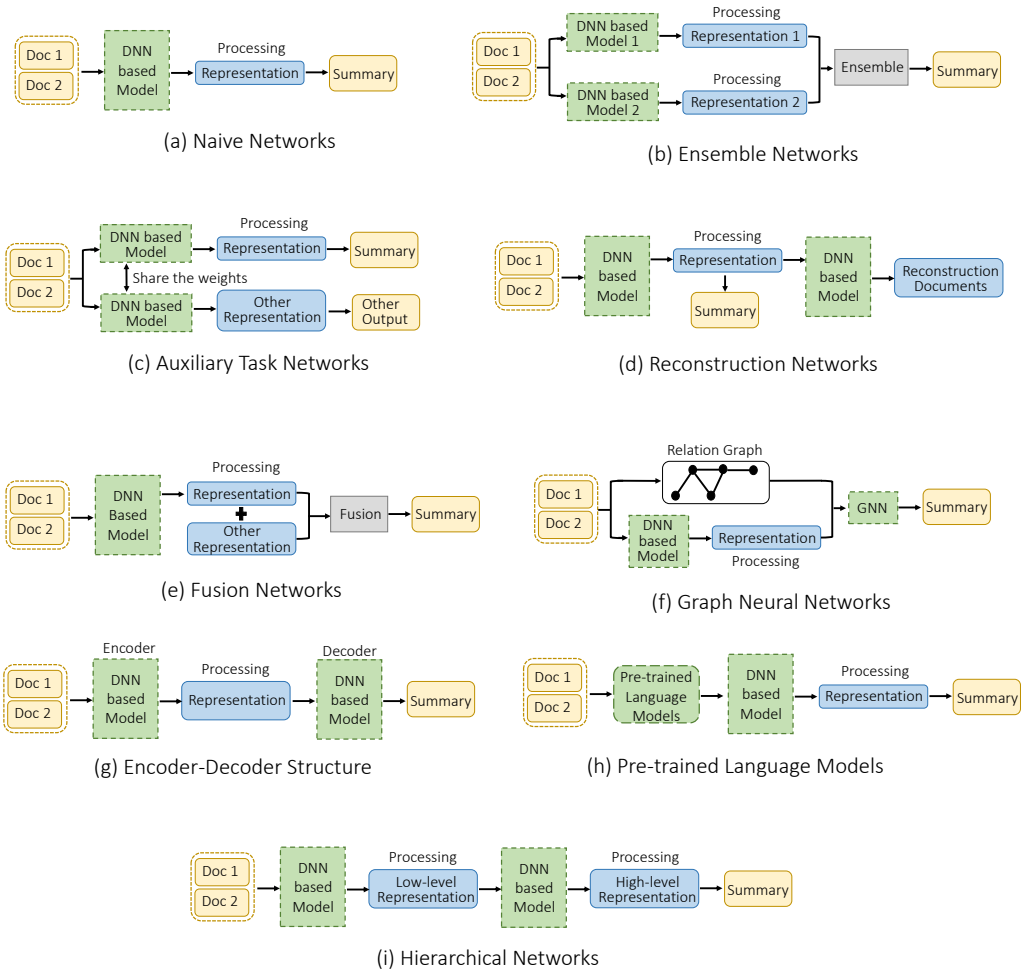


FIGURE 2.4. Network design strategies.

2.2.1 Architecture Design Strategies

Architecture design strategies play a critical role in deep learning based models, and many architectures have been applied to variants multi-document summarization tasks. Here, we generalized the network architectures and summarized them into nine types based on how they generate or fuse semantic-rich and syntactic-rich representation to improve multi-document summarization model performance (Figure 2.4); these different architectures can also be used as basic structures or stacked on each other to obtain more diverse design strategies. In Figure 2.4, deep neural models are in green boxes and can be flexibly substituted with other backbone networks. The blue boxes indicate the neural embeddings processed by neural networks or heuristic-designed approaches, e.g., "sentence/document" or "other" representation. The explanation for each sub-figure is listed as follows:

- *Naive Networks (Figure 2.4(a)).* Multiple concatenated documents are input through DNN based models to extract features. Word-level, sentence-level or

document-level representation is used to generate the downstream summary or select sentences. Naive networks represent the most naive model that lays the foundation for other strategies.

- *Ensemble Networks (Figure 2.4(b)).* Ensemble based methods leverage multiple learning algorithms to obtain better performance than individual algorithms. To capture semantic-rich and syntactic-rich representation, ensemble networks feed input documents to multiple paths with different network structures or operations. Later on, the representation from different networks is fused to enhance model expression capability. The majority vote or the average score can be used to determine the final output.

- *Auxiliary Task Networks (Figure 2.4(c))* employ different tasks in the summarization models, where text classification, text reconstruction, or other auxiliary tasks serve as complementary representation learners to obtain advanced features. Meanwhile, auxiliary task networks also provide researchers with a solution to use appropriate data from other tasks. In this strategy, parameter sharing schemes are used for jointly optimizing different tasks.

- *Reconstruction Networks (Figure 2.4(d))* optimize models from an unsupervised learning paradigm, which allows summarization models to overcome the limitation of insufficient annotated gold summaries. The use of such a paradigm enables generated summaries to be constrained in the natural language domain in a good manner.

- *Fusion Networks (Figure 2.4(e))* fuse representation generated from neural networks and hand-crafted features. These hand-crafted features contain adequate prior knowledge that facilitates the optimization of summarization models.

- *Graph Neural Networks (Figure 2.4(f)).* This strategy captures cross-document relations, crucial and beneficial for multi-document model training, by constructing graph structures based on the source documents, including word, sentence, or document-level information.

- *Encoder-Decoder Structure (Figure 2.4(g)).* The encoder embeds source documents into the hidden representation, i.e., word, sentence and document representation. This representation, containing compressed semantic and syntactic information, is passed to the decoder which processes the latent embeddings to synthesize local and global semantic/syntactic information to produce the final summaries.

- *Pre-trained Language Models (Figure 2.4(h))* obtain contextualized text representation by predicting words or phrases based on their context using large amounts of the corpus, which can be further fine-tuned for downstream task adaption (Dong et al., 2019). The models can fine-tune with randomly initialized decoders in an end-to-end fashion since transfer learning can assist the model training process (Li et al., 2020b).

- *Hierarchical Networks (Figure 2.4(i)).* Multiple documents are concatenated as inputs to feed into the first DNN based model to capture low-level representation. Another DNN based model is cascaded to generate high-level representation based on the previous ones. The hierarchical networks empower the model with the ability to capture abstract-level and semantic-level features more efficiently.

2.2.2 Recurrent Neural Networks based Models

Recurrent Neural Networks (RNNs) (Rumelhart, Hinton, and Williams, 1986) excel in modeling sequential data by capturing sequential relations and syntactic/semantic information from word sequences. In RNN models, neurons are connected through hidden layers and unlike other neural network structures, the inputs of each RNN neuron come not only from the word or sentence embedding but also from the output of the previous hidden state. Despite being powerful, vanilla RNN models often encounter gradient explosion or vanishing issues, so a large number of RNN-variants have been proposed. The most prevalent ones are Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), Gated Recurrent Unit (GRU) (Chung et al., 2014) and Bi-directional Long Short-Term Memory (Bi-LSTM) (Huang, Xu, and Yu, 2015). The DNN based Model in Figure 2.4 can be replaced with RNN based models to design models.

RNN based models have been used in multi-document summarization tasks since 2015. Cao et al. (Cao et al., 2015a) proposed an RNN-based model termed *Ranking framework upon Recursive Neural Networks (R2N2)*, which leverages manually extracted words and sentence-level features as inputs. This model transfers the sentence ranking task into a hierarchical regression process, which measures the importance of sentences and constituents in the parsing tree. Zheng et al. (Zheng et al., 2019) used a hierarchical RNN structure to utilize the subtopic information by extracting not only sentence and document embeddings, but also topic embeddings. In this SubTopic-Driven Summarization *(STDS)* model, the readers' comments are seen as auxiliary documents and the model employs soft clustering to incorporate comment and sentence representation for further obtaining subtopic representation. Arthur et

al. (Bražinskas, Lapata, and Titov, 2019) introduced a GRU-based encoder-decoder architecture to minimize the diversity of opinions reflecting the dominant views while generating multi-review summaries. Mao et al. (Mao et al., 2020) proposed a maximal margin relevance guided reinforcement learning framework (RL-MMR) to incorporate the advantages of neural sequence learning and statistical measures. The proposed soft attention for learning adequate representation allows more exploration of search space.

To leverage the advantage of the hybrid summarization model, Reinald et al. (Amplayo and Lapata, 2021) proposed a two-stage framework, viewing opinion summarization as an instance of multi-source transduction to distill salient information from source documents. The first stage of the model leverages a Bi-LSTM auto-encoder to learn word and document-level representation; the second stage fuses multi-source representation and generates an opinion summary with a simple LSTM decoder combined with a vanilla attention mechanism (Bahdanau, Cho, and Bengio, 2015) and a copy mechanism (Vinyals, Fortunato, and Jaitly, 2015).

Since paired multi-document summarization datasets are rare and hard to obtain, Li et al. (Li et al., 2017b) developed a RNN-based framework to extract salient information vectors from sentences in input documents in an unsupervised manner. Cascaded attention retains the most relevant embeddings to reconstruct the original input sentence vectors. During the reconstruction process, the proposed model leverages a sparsity constraint to penalize trivial information in the output vectors. Also, Chu et al. (Chu and Liu, 2019) proposed an unsupervised end-to-end abstractive summarization architecture called *MeanSum*. This LSTM-based model formalizes product or business reviews summarization problem into two individual closed-loops. Inspired by MeanSum, Coavoux et al. (Coavoux, Elsahar, and Gallé, 2019) used a two-layer standard LSTM to construct sentence representation for aspect-based multi-document abstractive summarization, and discovered that the clustering strategy empowers the model to reward review diversity and handle contradictory ones.

2.2.3 Convolutional Neural Networks Based Models

Convolutional neural networks (CNNs) (LeCun et al., 1998) achieve excellent results in computer vision tasks. The convolution operation scans through the word/sentence embeddings and uses convolution kernels to extract important information from input data objects. Using a pooling operation at intervals can return simple to complex feature levels. CNNs have been proven to be effective for various NLP tasks in recent years (Kim, 2014; Dos Santos and Gatti, 2014) as they can process natural language after sentence/word vectorization. Most of the CNN based multi-document

summarization models use CNNs for semantic and syntactic feature representation. As with RNN, CNN-based models can also replace DNN-based models in network design strategies (Please refer to Figure 2.4).

A simple way to use CNNs in multi-document summarization is by sliding multiple filters with different window sizes over the input documents for semantic representation. Cao et al. (Cao et al., 2015b) proposed a hybrid CNN-based model *PriorSum* to capture latent document representation. The proposed representation learner slides over the input documents with filters of different window widths and two-layer max-over-time pooling operations (Collobert et al., 2011) to fetch document-independent features that are more informative than using standard CNNs. Similarly, *HNet* (Singh, Gupta, and Varma, 2018) uses distinct CNN filters and max-over-time-pooling to generate salient feature representation for downstream processes. Cho et al. (Cho et al., 2019) also used different filter sizes in *DPP-combined* model to extract low-level features. Yin et al. (Yin and Pei, 2015) presented an unsupervised CNN-based model termed *Novel Neural Language Model (NNLM)* to extract sentence representation and diminish the redundancy of sentence selection. The NNLM framework contains only one convolution layer and one max-pooling layer, and both element-wise averaging sentence representation and context words representation are used to predict the next word. For aspect-based opinion summarization, Stefanos et al. (Angelidis and Lapata, 2018) leveraged a CNN based model to encode the product reviews which contain a set of segments for opinion polarity.

People with different background knowledge and understanding can produce different summaries of the same documents. To account for this variability, Zhang et al. (Zhang et al., 2016) suggested a *MV-CNN* model that ensembles three individual models to incorporate multi-view learning and CNNs to improve the performance of multi-document summarization. In this work, three CNNs with dual-convolutional layers used multiple filters with different window sizes to extract distinct saliency scores of sentences.

To overcome the insufficient training data problem, Cao et al. (Cao et al., 2017) developed a *TCSum* model incorporating an auxiliary text classification sub-task into multi-document summarization to introduce more supervision signals. The text classification model uses a CNN descriptor to project documents onto the distributed representation and to classify input documents into different categories. The summarization model shares the projected sentence embedding from the classification model, and the TCSum model then chooses the corresponding category based transformation matrices according to classification results to transform the sentence embedding into the summary embedding.

Unlike RNNs that support the processing of long time-serial signals, a naive CNN

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

2.2.4 Graph Neural Networks Based Models

CNNs have been successfully applied to many computer vision tasks to extract distinguished image features from the Euclidean space, but struggle when processing non-Euclidean data. Natural language data consist of vocabularies and phrases with strong relations which can be better represented with graphs than with sequential orders. Graph neural networks (GNNs, Figure 2.4 (f)) are composed of an ideal architecture for NLP since they can model strong relations between entities semantically and syntactically. Graph convolution networks (GCNs) and graph attention networks (GANs) are the most commonly adopted GNNs because of their efficiency and simplicity for integration with other neural networks. These models first build a relation graph based on input documents, where nodes can be words, sentences or documents, and edges capture the similarity among them. At the same time, input documents are fed into a DNN based model to generate embeddings at different levels. The GNNs are then built over the top to capture salient contextual information.

Yasunage et al. (Yasunaga et al., 2017) developed a GCN based extractive model to capture the relations between sentences. This model first builds a sentence-based graph and then feeds the pre-processed data into a GCN (Kipf and Welling, 2017) to capture sentence-wise related features. Defined by the model, each sentence is regarded as a node and the relation between each pair of sentences is defined as an edge. Inside each document cluster, the sentence relation graph can be generated through a cosine similarity graph (Erkan and Radev, 2004a), approximate discourse graph (Christensen, Soderland, Etzioni, et al., 2013), and the proposed personalized discourse graph. Both the sentence relation graph and sentence embeddings extracted by a sentence-level RNN are fed into GCN to produce the final sentence representation. With the help of a document-level GRU, the model generates cluster embeddings to fully aggregate features between sentences.

Similarly, Antognini et al. (Antognini and Faltings, 2019) proposed a GCN based model named *SemSentSum* that constructs a graph based on sentence relations. In contrast to Yasunage et al. (Yasunaga et al., 2017), this work leverages external

universal embeddings, pre-trained on the unrelated corpus, to construct a sentence semantic relation graph. Additionally, an edge removal method has been applied to deal with the sparse graph problems emphasizing high sentence similarities; if the weight of the edge is lower than a given threshold, the edge is removed. The sentence relation graph and sentence embeddings are fed into a GCN (Kipf and Welling, 2017) to generate saliency estimation for extractive summaries.

Yasunage et al. (Yasunaga et al., 2019) also designed a GCN based model for summarizing scientific papers. The proposed *ScisummNet* model uses not only the abstract of source scientific papers but also the relevant text from papers that cite the original source. The total number of citations is also incorporated into the model as an authority feature. A cosine similarity graph is applied to form the sentence relation graph, and GCNs are adopted to predict the sentence salience estimation from the sentence relation graph, authority scores and sentence embeddings.

Existing GNN based models focused mainly on the relationships between sentences, and do not fully consider the relationships between words, sentences, and documents. To fill this gap, Wang et al. (Wang et al., 2020a) proposed a heterogeneous GAN based model, called *HeterDoc-SUM Graph*, that is specific for extractive multi-document summarization. This heterogeneous graph structure includes word, sentence, and document nodes, where sentence nodes and document nodes are connected according to the contained word nodes. Word nodes thus act as an intermediate bridge to connect the sentence and document nodes, and are used to better establish document-document, sentence-sentence and sentence-document relations. TF-IDF values are used to weight word-sentence and word-document edges, and the node representation of these three levels are passed into the graph attention networks for model update. In each iteration, bi-directional updating of both word-sentence and word-document relations are performed to better aggregate cross-level semantic knowledge.

2.2.5 Pointer-generator Networks Based Models

Pointer-generator (PG) networks (See, Liu, and Manning, 2017a) are proposed to overcome the problems of factual errors and high redundancy in the summarization tasks. This network has been inspired by Pointer Network (Vinyals, Fortunato, and Jaitly, 2015), CopyNet (Gu et al., 2016), forced-attention sentence compression (Miao and Blunsom, 2016), and coverage mechanism from machine translation (Tu et al., 2016). PG networks combine sequence-to-sequence model and pointer networks to obtain a united probability distribution allowing vocabularies to be selected from source texts or generated by machines. Additionally, the coverage mechanism prevents PG networks from consistently choosing the same phrases.

The *Maximal Marginal Relevance (MMR)* method is designed to select a set of salient sentences from source documents by considering both *importance* and *redundancy* indices (Carbonell and Goldstein, 1998a). The redundancy score controls sentence selection to minimize overlap with the existing summary. The MMR model adds a new sentence to the objective summary based on importance and redundancy scores until the summary length reaches a certain threshold. Inspired by MMR, Alexander et al. (Fabbri et al., 2019b) proposed an end-to-end *Hierarchical MMR-Attention Pointer-generator (Hi-MAP)* model to incorporate PG networks and MMR (Carbonell and Goldstein, 1998a) for abstractive multi-document summarization. The Hi-MAP model improves PG networks by modifying attention weights (multiplying MMR scores by the original attention weights) to include better important sentences in, and filter redundant information from, the summary. Similarly, the MMR approach is implemented by *PG-MMR* model (Lebanoff, Song, and Liu, 2018) to identify salient source sentences from multi-document inputs, albeit with a different method for calculating MMR scores from Hi-MAP; instead, ROUGE-L Recall and ROUGE-L Precision (Lin, 2004a) serve as evaluation metrics to calculate the importance and redundancy scores. To overcome the scarcity of multi-document summarization datasets, the PG-MMR model leverages a support vector regression model that is pre-trained on a single document summarization dataset to recognize the important contents. This support vector regression model also calculates the score of each input sentence by considering four factors: sentence length, sentence relative/absolute position, sentence-document similarities, and sentence quality obtained by a PG network. Sentences with the top-K scores are fed into another PG network to generate a concise summary.

2.2.6 Transformer Based Models

As discussed, CNN based models are not as good at processing sequential data as RNN based models. However, RNN based models are not amenable to parallel computing, as the current states in RNN models highly depend on results from the previous steps. Additionally, RNNs struggle to process long sequences since former knowledge will fade away during the learning process. Adopting *Transformer* based architectures (Vaswani et al., 2017) is one solution to solve these problems. The Transformer is based on the self-attention mechanism, has natural advantages for parallelization, and retains relative long-range dependencies. The Transformer model has achieved promising results in multi-document summarization tasks (Liu et al., 2018a; Liu and Lapata, 2019a; Li et al., 2020b; Jin, Wang, and Wan, 2020a; Chen et al., 2021a; Xiao et al., 2022; Moro et al., 2022; Wen et al., 2022) and can replace the *DNN based Model* in Figure 2.4. Most of the Transformer based models

follow an encoder-decoder structure. Transformer based models can be divided into flat Transformer, hierarchical Transformer, and pre-train language models.

Flat Transformer. Liu et al. (Liu et al., 2018a) introduced Transformer to multi-document summarization tasks, aiming to generate a Wikipedia article from a given topic and set of references. The model selects a series of top-K tokens and feeds them into a Transformer based decoder-only sequence transduction model to generate Wikipedia articles. More specifically, the Transformer decoder-only architecture combines the results from the extractive stage and gold summary into a sentence for training. To obtain rich semantic representation from different granularity, Jin et al. (Jin, Wang, and Wan, 2020a) proposed a Transformer based multi-granularity interaction network *MGSum* and unified extractive and abstractive multi-document summarization. Words, sentences, and documents are considered as three granular levels of semantic unit connected by a granularity hierarchical relation graph. In the same granularity, a self-attention mechanism is used to capture the semantic relationships. Sentence granularity representation is employed in the extractive summarization, and word granularity representation is adapted to generate an abstractive summary. Brazinskas et al. (Brazinskas, Lapata, and Titov, 2020) created a precedent for few-shot learning for multi-document summarization that leverages a Transformer conditional language model and a plug-in network for both extractive and abstractive multi-document summarization to overcome rapid overfitting and poor generation problems resulting from naive fine-tuning of large parameter models.

Hierarchical Transformer. To handle huge amounts of input documents (currently many large scale multi-document summarization datasets contain more than ten thousand input document sets), Yang et al. (Liu and Lapata, 2019a) proposed a two-stage *Hierarchical Transformer (HT) model* with an inter-paragraph and graph-informed attention mechanism that allows the model to encode multiple input documents hierarchically instead of by simple flat-concatenation. A logistic regression model is employed to select the top-K paragraphs, which are fed into a local Transformer layer to obtain contextual features. A global Transformer layer mixes the contextual information to model the dependencies of the selected paragraphs. To leverage graph structure, Chen et al. (Chen et al., 2021a) used a hierarchical Transformer to encode the document graph and select the summary sub-graph. The document graph is a directed acyclic graph that represents the relations between sentences in multiple documents. The summary sub-graph is a sub-DAG that contains the most salient and relevant sentences for the summary. Another work to leverage graph information is model *GraphSum*, an end-to-end Transformer based model based on the HT model. In the graph encoding layers, GraphSum extends the self-attention mechanism to the graph-informed self-attention mechanism, which incorporates the

graph representation into the Transformer encoding process. Furthermore, the Gaussian function is applied to the graph representation matrix to control the intensity of the graph structure's impact on the summarization model. The HT and GraphSum models are both based on the self-attention mechanism leading quadratic memory growth increases with the number of input sequences; to address this issue, Pasunuru et al. (Pasunuru et al., 2021b) modified the full self-attention with local and global attention mechanism (Beltagy, Peters, and Cohan, 2020) to scale the memory linearly. Dual encoders are proposed for encoding truncated concatenated documents and linearized graph information from full documents.

Pre-trained language models (LMs). Pre-trained Transformers on large text corpora have shown great successes in downstream NLP tasks including text summarization. The pre-trained LMs can be trained on non-summarization or single document summarization datasets to overcome lack of multi-document summarization data (Zhang et al., 2020a; Li et al., 2020b; Pasunuru et al., 2021b), which helps to improve the model performance. In hierarchical Transformer architecture, replacing the low-level Transformer (token-level) encoding layer with pre-trained LMs helps the model breakthrough length limitations to perceive further information (Li et al., 2020b). Inside a hierarchical Transformer architecture, the output vector of the "[CLS]" token can be used as input for high-level Transformer models. To avoid the self-attention quadratic-memory increment when dealing with document-scale sequences, a Longformer based approach (Beltagy, Peters, and Cohan, 2020), including local and global attention mechanisms, can be incorporated with pre-trained LMs to scale the memory linearly for multi-document summarization (Pasunuru et al., 2021b). Another solution for computational issues can be borrowed from single document summarization is to use a multi-layer Transformer architecture to scale the length of documents allowing pre-trained LMs to encode a small block of text and the information can be shared among the blocks between two successive layers (Grail, Perez, and Gaussier, 2021). BART (Lewis et al., 2020), GPT-2 (Radford et al., 2019), and T5 (Raffel et al., 2020) are pre-trained language models that can be used for language generation and they have been applied for multi-document summarization tasks (Pang et al., 2021; Su et al., 2020; Alambo et al., 2020; Moro et al., 2022; Wen et al., 2022). Instead of regular language models, PEGASUS (Zhang et al., 2020a) is a pre-trained Transformer-based encoder-decoder model with gap-sentences generation (GSG) that focused on abstractive summarization. GSG shows that masking whole sentences based on importance, instead of through random or lead selection, works well for downstream summarization tasks. BART, T5, and PEGASUS are based on data-rich single document summarization settings. Goodwin et al. (Goodwin, Savery, and Demner-Fushman, 2020) evaluated these three pre-trained models

on four multi-document summarization datasets and suggested that while large improvements have been made on the standard single document summarization task, highly abstractive multi-document summarization remains a challenge. PRIMERA (Xiao et al., 2022) is a pre-trained model specifically designed for multi-document summarization which can serve as a zero-shot summarizer.

2.2.7 Deep Hybrid Models

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

CNN + LSTM + Capsule networks. Cho et al. (Cho et al., 2019) proposed a hybrid model based on the determinantal point processes for semantically measuring sentence similarities. A convolutional layer slides over the pairwise sentences with filters of different sizes to extract low-level features. Capsule networks (Sabour, Frosst, and Hinton, 2017; Yang et al., 2018) are employed to identify redundant information by transforming the spatial and orientational relationships for high-level representation. The authors also used LSTM to reconstruct pairwise sentences and add reconstruction loss to the final objective function.

CNN + Bi-LSTM + Multi-layer Perceptron (MLP). Abhishek et al. (Singh, Gupta, and Varma, 2018) proposed an extractive MDS framework that considers document-dependent and document-independent information. In this model, a CNN with different filters captures phrase-level representation. Full binary trees formed with these salient representation are fed to the recommended Bi-LSTM tree indexer to enable better generalization abilities. A MLP with ReLU function is employed for leaf node transformation. More specifically, the Bi-LSTM tree indexer leverages the time serial power of LSTMs and the compositionality of recursive models to capture both semantic and compositional features.

PG networks + Transformer. In generating a summary, it is necessary to consider the information fusion of multiple sentences, especially sentence pairs. Logan et al. (Lebanoff et al., 2019) found the majority of summary sentences are generated by fusing one or two source sentences; so they proposed a two-stage summarization method that considers the semantic compatibility of sentence pairs. This method joint-scores single sentence and sentence pairs to filter representative from the original documents. Sentences or sentence pairs with high scores are then compressed and rewritten to generate a summary that leverages PG network. 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.

2.2.8 The Variants of Multi-document Summarization

In this section, we briefly introduced several multi-document summarization task variants which can be modeled as multi-document summarization problems and adopt the aforementioned deep learning techniques and neural network architectures.

Query-oriented MDS calls for a summary from a set of documents that answers a query. It tries to solve realistic query-oriented scenario problems and only summarizes important information that best answers the query in a logical order (Pasunuru et al., 2021a). Specifically, query-oriented multi-document summarization combines the information retrieval and multi-document summarization techniques. The content that needs to be summarized is based on the given queries. Liu et al. (Liu and Lapata, 2019a) incorporated the query by simply prepending the query to the top-ranked document during encoding. Pasunuru (Pasunuru et al., 2021a) involved a query encoder and integrated query embedding into an multi-document summarization model, ranking the importance of documents for a given query.

Dialogue summarization aims to provide a succinct synopsis from multiple textual utterances of two or more participants, which could help quickly capture relevant information without having to listen to long and convoluted dialogues (Liu et al., 2019). Dialogue summary covers several areas, including meetings (Zhu et al., 2020; Koay et al., 2020; Feng et al., 2021), email threads (Zhang et al., 2021), medical dialogues (Song et al., 2020b; Joshi et al., 2020; Enarvi et al., 2020), customer service (Liu et al., 2019) and media interviews (Zhu et al., 2021). Challenges in dialogue summarization can be summarized into the following seven categories: informal language use, multiple participants, multiple turns, referral and coreference, repetition and interruption, negations and rhetorical questions, role and language change (Chen and Yang, 2020). The flow of the dialogue would be neglected if multi-document summarization models are directly applied for dialogue summarization. Liu et al. (Liu et al., 2019) relied on human annotations to capture the logic of the dialogue. Wu et al. (Wu et al., 2021) used summary sketch to identify the interaction between speakers and their corresponding textual utterances in each turn. Chen et al. (Chen and Yang, 2020) proposed a multi-view sequence to sequence based encoder to extract dialogue structure and a multi-view decoder to incorporate different views to generate final summaries.

Stream summarization aims to summarize new documents in a continuously growing document stream, such as information from social media. Temporal summarization and real-time summarization (RTS)[1] can be seen as a form of stream document summarization. Stream summarization considers both historical dependencies and

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

future uncertainty of the document stream. Yang et al. (Yang et al., 2020) used deep reinforcement learning to solve the relevance, redundancy, and timeliness issues in steam summarization. Tan et al. (Tan, Lu, and Li, 2017) transformed the real time summarization task as a sequential decision-making problem and used a LSTM layer and three fully connected neural network layers to maximize the long-term rewards.

2.3 Multi-document Summarization Objective Functions

In this section, we will take a closer look at different objective functions adopted by various multi-document summarization models. In summarization models, objective functions play an important role by guiding the model to achieve specific purposes.

2.3.1 Cross-Entropy Objective

Cross-entropy usually acts as an objective function to measure the distance between two distributions. Many existing multi-document summarization models adopt it to measure the difference between the distributions of generated summaries and the gold summaries (Cao et al., 2015a; Zhang et al., 2016; Wang et al., 2020a; Zhang, Tan, and Wan, 2018; Cho et al., 2019; Yasunaga et al., 2019). Formally, the cross-entropy loss is defined as:

$$L_{CE} = -\sum_{i=1} \mathbf{y_i} \log(\hat{\mathbf{y}}_\mathbf{i}), \tag{2.1}$$

where $\mathbf{y_i}$ is the target score from gold summaries and machine-generated summaries, and $\hat{\mathbf{y}}_\mathbf{i}$ is the predicted estimation from the deep learning based models. Different from calculations in other tasks, such as text classification, in summarization tasks, $\mathbf{y_i}$ and $\hat{\mathbf{y}}_\mathbf{i}$ have several methods to calculate. $\hat{\mathbf{y}}_\mathbf{i}$ usually is calculated by Recall-Oriented Understudy for Gisting Evaluation (ROUGE). For example, ROUGE-1(Antognini and Faltings, 2019), ROUGE-2 (Liu and Lapata, 2019a) or the normalized average of ROUGE-1 and ROUGE-2 scores (Yasunaga et al., 2017) could be adopted to compute the ground truth score between the selected sentences and gold summary.

2.3.2 Reconstructive Objective

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

$$L_{Rec} = \left\| \mathbf{x_i} - \phi'(\phi(\mathbf{x_i}; \theta); \theta') \right\|_* , \tag{2.2}$$

where $\mathbf{x_i}$ represents the input vector; ϕ and ϕ' represent the encoder and decoder with θ and θ' as their parameters respectively, $||\cdot||_*$ represents norm (* stands for 0, 1, 2, ..., infinity). L_{Rec} is a measuring function to calculate the distance between source documents and their reconstructive outputs. Chu et al. (Chu and Liu, 2019) used a reconstructive loss to constrain the generated text into the natural language domain, reconstructing reviews in a token-by-token manner. Moreover, this paper also proposes a variant termed *reconstruction cycle loss*. By using the variant, the reviews are encoded into a latent space to further generate the summary, and the summary is then decoded to the reconstructed reviews to form another reconstructive closed-loop. An unsupervised learning loss was designed by Li et al. (Li et al., 2017b) to reconstruct the condensed output vectors to the original input sentence vectors with L_2 distance. This paper further constrains the condensed output vector with a L_1 regularizer to ensure sparsity. Similarly, Zheng et al. (Zheng et al., 2019) adopted a bi-directional GRU encoder-decoder framework to reconstruct both news and comment sentences in a word sequence manner. Liu et al. (Liu et al., 2018a) concatenated both input and output sequences to predict the next token to train the abstractive model. There are also some variants, such as leveraging the latent vectors of variational auto-encoder for reconstruction to capture better representation. Li et al. (Li, Bing, and Lam, 2017) introduced three individual reconstructive losses to consider both news reconstruction and comments reconstruction separately, along with a variational auto-encoder lower bound. Bravzinskas et al. (Bražinskas, Lapata, and Titov, 2019) utilized a variational auto-encoder to generate the latent vectors of given reviews, where each review is reconstructed by the latent vectors combined with other reviews.

2.3.3 Redundancy Objective

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

$$L_{Red} = Sim(\mathbf{x_i}, \mathbf{x_j}), \tag{2.3}$$

where $Sim(\cdot)$ is the similarity function to measure the overlap between different $\mathbf{x_i}$ and $\mathbf{x_j}$, which can be phrases, sentences, topics or documents. The redundancy objective is often treated as an auxiliary objective combined with other loss functions. Li et al. (Li et al., 2017b) penalized phrase pairs with similar meanings to eliminate the redundancy. Nayeem et al. (Nayeem, Fuad, and Chali, 2018) used the redundancy objective to avoid generating repetitive phrases, constraining a sentence to appear only once while maximizing the scores of important phrases. Zheng et al. (Zheng et

al., 2019) adopted a redundancy loss function to measure overlaps between subtopics; intuitively, smaller overlaps between subtopics resulted in less redundancy in the output domain. Yin et al. (Yin and Pei, 2015) proposed a redundancy objective to estimate the diversity between different sentences.

2.3.4 Max Margin Objective

Max Margin Objectives (MMO) are also used to empower the multi-document summarization models to learn better representation. The objective function is formalized as:

$$L_{Margin} = \max\left(0, f(\mathbf{x_i};\theta) - f(\mathbf{x_j};\theta) + \gamma\right), \tag{2.4}$$

where $\mathbf{x_i}$ and $\mathbf{x_j}$ represent the input vectors, θ are parameters of the model function $f(\cdot)$, and γ is the margin threshold. The MMO aims to force function $f(\mathbf{x_i};\theta)$ and function $f(\mathbf{x_j};\theta)$ to be separated by a predefined margin γ. In Cao et al. (Cao et al., 2017), a MMO is designed to constrain a pair of randomly sampled sentences with different salience scores – the one with a higher score should be larger than the other one more than a marginal threshold. Two max margin losses are proposed in Zhong et al. (Zhong et al., 2020): a margin-based triplet loss that encouraged the model to pull the gold summaries semantically closer to the original documents than to the machine-generated summaries; and a pair-wise margin loss based on a greater margin between paired candidates with more disparate ROUGE score rankings.

2.3.5 Multi-Task Objective

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

$$L_{Mul} = L_{Summ} + L_{Other}, \tag{2.5}$$

where L_{Summ} is the loss function of multi-document summarization tasks, and L_{Other} is the loss function of an auxiliary task. Angelidis et al. (Angelidis and Lapata, 2018) assumed that the aspect-relevant words not only provide a reasonable basis for model aspect reconstruction, but also a good indicator for product domain. Similarly, multi-task classification was introduced by Cao et al. (Cao et al., 2017). Two models are maintained: text classification and text summarization models. In the first model, CNN is used to classify text categories and cross-entropy loss is used as the objective function. The summarization model and the text classification model share parameters and pooling operations, so are equivalent to the shared document vector

representation. Coavoux et al. (Coavoux, Elsahar, and Gallé, 2019) jointly optimized the model from a language modeling objective and two other multi-task supervised classification losses, which are polarity loss and aspect loss.

2.3.6 Other Types of Objectives

There are many other types of objectives in addition to those mentioned above. Cao et al. (Cao et al., 2015b) proposed using ROUGE-2 to calculate the sentence saliency scores and the model tries to estimate this saliency with linear regression. Yin et al. (Yin and Pei, 2015) suggested summing the squares of the prestige vectors calculated by the PageRank algorithm to identify sentence importance. Zhang et al. (Zhang et al., 2016) proposed an objective function by ensembling individual scores from multiple CNN models; besides the cross-entropy loss, a consensus objective is adopted to minimize disagreement between each pair of classifiers. Amplay et al. (Amplayo and Lapata, 2021) used two objectives in the abstract module: the first to optimize the generation probability distribution by maximizing the likelihood; and the second to constrain the model output to be close to its gold summary in the encoding space, as well as being distant from the random sampled negative summaries. Chu et al. (Chu and Liu, 2019) designed a similarity objective that shares the encoder and decoder weights within the auto-encoder module, while in the summarization module, the average cosine distance indicates the similarity between the generated summary and the reviews. A variant similarity objective termed *early cosine objective* is further proposed to compute the similarity in a latent space which is the average of the cell states and hidden states to constrain the generated summaries semantically close to reviews.

2.4 Multi-document Summarization Evaluation Metrics

Evaluation metrics are used to measure the effectiveness of a given method objectively, so well-defined evaluation metrics are crucial to multi-document summarization research. We classified the existing evaluation metrics into two categories and will discuss each category in detail: (1) ROUGE: the most commonly used evaluation metrics in the summarization community; and (2) other evaluation metrics that have not been widely used in multi-document summarization research to date. We summarize the advantages and disadvantages of above-mentioned evaluation metrics in Table 2.1.

TABLE 2.1. Advantages and disadvantages of different evaluation metrics.

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

2.4.1 ROUGE

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) (Lin, 2004a) is a collection of evaluation indicators that is one of the most essential metrics for many natural language processing tasks, including machine translation and text summarization. ROUGE obtains prediction/ground-truth similarity scores through comparing automatically generated summaries with a set of corresponding human-written references. ROUGE has many variants to measure candidate abstracts in a variety of ways (Lin, 2004a). *ROUGE-N* measures a n-gram recall between reference and their corresponding candidate summaries (Lin, 2004a). *ROUGE-L* adopts the longest common subsequence algorithm to count the longest matching vocabularies

(Lin, 2004a). *ROUGE-W* (Lin, 2004a) is proposed to weight consecutive matches to better measure semantic similarities between two texts. *ROUGE-S* (Lin, 2004a) stands for ROUGE with Skip-bigram co-occurrence statistics that allows the bigram to skip arbitrary words. An extension of ROUGE-S, *ROUGE-SU* (Lin, 2004a) refers to ROUGE with Skip-bigram plus Unigram-based co-occurrence statistics and is able to be obtained from ROUGE-S by adding a begin-of-sentence token at the start of both references and candidates. *ROUGE-WE* (Ng and Abrecht, 2015) is proposed to further extend ROUGE by measuring the pair-wise summary distances in word embedding space. In recent years, more ROUGE-based evaluation models have been proposed to compare gold and machine-generated summaries, not just according to their literal similarity, but also considering semantic similarity (ShafieiBavani et al., 2018; Zhao et al., 2019; Zhang et al., 2020b). In terms of the ROUGE metric for multiple gold summaries, the Jackknifing procedure (similar to K-fold validation) has been introduced (Lin, 2004a). The M best scores are computed from sets composed of M-1 reference summaries and the final ROUGE-N is the average of M scores. This procedure can also be applied to ROUGE-L, ROUGE-W and ROUGE-S.

2.4.2 Other Evaluation Metrics

Besides *ROUGE*-based (Lin, 2004a) metrics, other evaluation metrics for multi-document summarization exist, but have received less attention than ROUGE. Based on the mode of summaries matching, we divide the evaluation metrics into two groups: lexical matching metrics and semantic matching metrics.

Lexical Matching Metrics. *BLEU* (Papineni et al., 2002) is a commonly used vocabulary-based evaluation metric that provides a precision-based evaluation indicator, as opposed to ROUGE that mainly focuses on recall. *Perplexity* (Jelinek et al., 1977) is used to evaluate the quality of the language model by calculating the negative log probability of a word's appearance. A low perplexity on a test dataset is a strong indicator of a summary's high grammatical quality because it measures the probability of words appearing in sequences. Based on *Pyramid* (Nenkova, Passonneau, and McKeown, 2007) calculation, the abstract sentences are manually divided into several Summarization Content Units (SCUs), each representing a core concept formed from a single word or phrase/sentence. After sorting SCUs in order of importance to form the *Pyramid*, the quality of automatic summarization is evaluated by calculating the number and importance of SCUs included in the document (Nenkova and Passonneau, 2004). Intuitively, more important SCUs exist at higher levels of the pyramid. Although *Pyramid* shows a strong correlation with human judgment, it

requires professional annotations to match and evaluate SCUs in generated and gold summaries. Some recent works focus on the construction of *Pyramid* (Passonneau et al., 2013; Yang, Passonneau, and De Melo, 2016; Hirao, Kamigaito, and Nagata, 2018; Gao, Sun, and Passonneau, 2019; Shapira et al., 2019). *Responsiveness* (Louis and Nenkova, 2013) measures content selection and linguistic quality of summaries by directly rating scores. Additionally, the assessments are calculated without reference to model summaries. *Data Statistics* (Grusky, Naaman, and Artzi, 2018) contain three evaluation metrics: extractive fragment coverage measures the novelty of generated summaries by calculating the percentage of words in the summary that are also present in source documents; extractive fragment density measures the average length of the extractive block to which each word in the summary belongs; and compression ratio compares the word numbers in the source documents and generated summary.

Semantic Matching Metrics. *METEOR* (Metric for Evaluation of Translation with Explicit Ordering) (Banerjee and Lavie, 2005) is an improvement to BLEU. The main idea behind METEOR is that while candidate summaries can be correct with similar meanings, they are not exactly matched with references. In such a case, WordNet[2] is introduced to expand the synonym set, and the word form is also taken into account. *SUPERT* (Gao, Zhao, and Eger, 2020) is an unsupervised evaluation metric that measures the semantic similarity between the pseudo-reference summary and the machine-generated summary. *SUPERT* obviates the need for human annotations by not referring to gold summaries. Contextualized embeddings and soft token alignment techniques are leveraged to select salient information from the input documents to evaluate summary quality. *Preferences based Metric* (Zopf, 2018) is a pairwise sentence preference-based evaluation model and it does not depend on the gold summaries. The underlying premise is to ask annotators about their pair-wise preferences rather than writing complex gold summaries, and are much easier and faster to obtain than traditional reference summary-based evaluation models. *BERTScore* (Zhang et al., 2020b) computes a similarity score for each token within the candidate sentence and the reference sentence. It measures the soft overlap of two texts' BERT embeddings. *MoverScore* (Zhao et al., 2019) adopts a distance to evaluate the agreement between two texts in the context of BERT and ELMo word embeddings. This proposed metric has a high correlation with human judgment of text quality by adopting earth mover's distance. *Importance* (Peyrard, 2019) is a simple but rigorous evaluation metric from the aspect of information theory. It is a final indicator calculated from the three aspects: *Redundancy*, *Relevance*, and *Informativeness*. A good

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

TABLE 2.2. Comparison of different datasets. In the table, "Ave", "Summ", "Len", "bus","rev" and "#" represent average, summary, length, business, reviews and numbers respectively; "Docs" and "sents" mean documents and sentences respectively.

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	Wikipedia
Multi-News	-	train / val / test 44972 / 5622 / 5622 2-10 docs / cluster	1 summ / cluster	263.66 words 9.97 sents 262 tokens	News
Opinosis	51	6457 rev	5 summ / cluster	-	Site reviews
Rotten Tomatoes	3731	99.8 rev / cluster	1 summ / cluster	19.6 tokens	Movie reviews
Yelp	-	train / val / test bus: 10695 / 1337 / 1337 rev: 1038184 / 129856 / 129840	-	-	Customer reviews
Scisumm	1000	21 - 928 cites / paper 15 sents / refer	1 summ / cluster	151 words	Science Paper
WCEP	10200	235 docs / cluster	1 summ / cluster	32 words	Wikipedia
Multi-XScience	-	train / val / test 30369 / 5066 / 5093	1 summ / cluster	116.44 words	Science Paper

summary should have low *Redundancy* and high *Relevance* and high *Informativeness*. The cluster of *Human Evaluation* is used to supplement automatic evaluation on relatively small instances. Annotators evaluate the quality of machine-generated summaries by rating *Informativeness, Fluency, Conciseness, Readability, Relevance.* Model ratings are usually computed by averaging the rating on all selected summary pairs.

2.5 Multi-document Summarization Datasets

Compared to single document summarization tasks, large-scale multi-document summarization datasets, which contain more general scenarios with many downstream tasks, are relatively scarce. In this section, we presented our investigation on the 10 most representative datasets commonly used for multi-document summarization and its variant tasks. Table 2.2 compares the datasets based on the numbers of clusters

and documents; the number and the average length of summaries; and the field to which the dataset belongs.

DUC & TAC. DUC[3] (Document Understanding Conference) provides official text summarization competitions each year from 2001-2007 to promote summarization research. DUC changed its name to Text Analysis Conference (TAC)[4] in 2008. Here, the DUC datasets refer to the data collected from 2001-2007; the TAC datasets refer to the datasets after 2008. Both DUC and TAC are from the news domains, including various topics such as politics, natural disasters, and biography. Nevertheless, as shown in Table 2.2, the DUC and TAC datasets provide small datasets for model evaluation that only include hundreds of news documents and human-annotated summaries. Of note, the first sentence in a news item is usually information-rich that renders bias in the news datasets, so it fails to reflect the structure of natural documents in daily lives. These two datasets are on a relatively small scale and not ideal for large-scale deep neural based multi-document summarization model training and evaluation.

OPOSUM. OPOSUM (Angelidis and Lapata, 2018) collects multiple reviews of six product domains from Amazon. This dataset not only contains multiple reviews and corresponding summaries but also products' domain and polarity information. The latter information could be used as auxiliary supervision signals.

WikiSum. WikiSum (Liu et al., 2018a) targets abstractive multi-document summarization. For a specific Wikipedia theme, the documents cited in Wikipedia articles or the top-10 Google search results (using the Wikipedia theme as a query) are seen as the source documents. gold summaries are the real Wikipedia articles. However, some of the URLs are not available and can be identical to each other in parts. To remedy these problems, Liu et al. (Liu and Lapata, 2019a) cleaned the dataset and deleted duplicated examples, so here we report statistical results from (Liu and Lapata, 2019a).

Multi-News. Multi-News (Fabbri et al., 2019b) is a relatively large-scale dataset in the news domain; the articles and human-written summaries are all from the Web[5]. This dataset includes 56,216 article-summary pairs and contains trace-back links to the original documents. Moreover, the authors compared the Multi-News dataset with prior datasets in terms of coverage, density, and compression, revealing that this dataset has various arrangement styles of sequences.

Opinosis. The Opinosis dataset (Ganesan, Zhai, and Han, 2010) contains reviews

[3]http://duc.nist.gov/
[4]http://www.nist.gov/tac/
[5]http://newser.com

of 51 topic clusters collected from TripAdvisor[6], Amazon[7], and Edmunds[8]. For each topic, approximately 100 sentences on average are provided and the reviews are fetched from different sources. For each cluster, five professionally written gold summaries are provided for model training and evaluation.

Rotten Tomatoes. The Rotten Tomatoes dataset (Wang and Ling, 2016) consists of the collected reviews of 3,731 movies from the Rotten Tomato website[9]. The reviews contain both professional critics and user comments. For each movie, a one-sentence summary is created by professional editors.

Yelp. Chu et al. (Chu and Liu, 2019) proposed a dataset named Yelp based on the Yelp Dataset Challenge. This dataset includes multiple customer reviews with five-star ratings. The authors provided 100 manual-written summaries for model evaluation using Amazon Mechanical Turk (AMT), within which every eight input reviews are summarized into one gold summary.

Scisumm. Scisumm dataset (Yasunaga et al., 2019) is a large, manually annotated corpus for scientific document summarization. The input documents are a scientific publication, called the reference paper, and multiple sentences from the literature that cite this reference paper. In the SciSumm dataset, the 1,000 most cited papers from the ACL Anthology Network (Radev et al., 2013) are treated as reference papers, and an average of 15 citation sentences are provided after cleaning. For each cluster, one gold summary is created by five NLP-based Ph.D. students or equivalent professionals.

WCEP. The Wikipedia Current Events Portal dataset (WCEP) (Ghalandari et al., 2020a) contains human-written summaries of recent news events. Similar articles are provided by searching similar articles from Common Crawl News dataset[10] to extend the inputs to obtain large-scale news articles. Overall, the WCEP dataset has good alignment with real-world industrial use cases.

Multi-XScience. The source data of Multi-XScience (Lu, Dong, and Charlin, 2020a) are from Arxiv and Microsoft academic graphs and this dataset is suitable for abstractive multi-document summarization. Multi-XScience contains fewer positional and extractive biases than the WikiSum and Multi-News datasets, so the drawback of obtaining higher scores from a copy sentence at a certain position can be partially avoided.

[6]https://www.tripadvisor.com/
[7]https://www.amazon.com.au/
[8]https://www.edmunds.com/
[9]http://rottentomatoes.com
[10]https://commoncrawl.org/2016/10/news-dataset-available/

Datasets for MDS Variants. The representative query-oriented multi-document summarization datasets are Debatepedia (Nema et al., 2017), AQUAMUSE (Kulkarni et al., 2020), and QBSUM (Zhao et al., 2021). The representative dialogue summarization datasets are DIALOGSUM (Chen et al., 2021b), AMI (Carletta et al., 2005), MEDIASUM (Zhu et al., 2021), and QMSum (Zhong et al., 2021). RTS is a track at the Text Retrieval Conference (TREC) which provides several RTS datasets[11]. Tweet Contextualization track (Bellot et al., 2016) (2012-2014) is derived from the INEX 2011 Question Answering Track, that focuses on more NLP-oriented tasks and moves to multi-document summarization.

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

Chapter 3

Enhancing Abstractive Multi-document Summarization with Linguistic Knowledge

3.1 Introduction

Recent years have witnessed an increasing number of neural network models applied in MDS (Fabbri et al., 2019a; Liu and Lapata, 2019a; Li et al., 2020c; Jin, Wang, and Wan, 2020b; Xiao et al., 2022) due to the rapid improvement of computational power (Ma et al., 2020). Transformer (Vaswani et al., 2017) is a popular one among them. It is based on a self-attention mechanism and has natural advantages for parallelization and could retain long-range relations between pairs of tokens among documents. Liu et al. (Liu et al., 2018b) adopted a flat Transformer model to generate Wikipedia articles. The model selects top-K tokens and feeds them into the decoder-only sequence transduction. Besides Flat Transformer, Hierarchical Transformer-based models (Liu and Lapata, 2019a; Li et al., 2020d; Pasunuru et al., 2021b) utilize multiple encoders to embed the hierarchical relations among the source documents. Wen et al. (Xiao et al., 2022) proposed a pre-train language model PRIMERA, using encoder-decoder transformers to simplify the processing of concatenated input documents, leverages the Longformer (Beltagy, Peters, and Cohan, 2020) to pre-train with a novel entity-based sentence masking objective. However, computing token-wise self-attention in the Transformer takes pairs of token relations into account but lacks syntactic support that may cause content irrelevance and deviation for summary generation (Jin, Wang, and Wan, 2020c).

Many research works seek to incorporate linguistic knowledge to further improve the quality of summaries. Daniel et al. (Leite et al., 2007) suggested that linguistic knowledge help improve the informativeness of summaries. Sho et al. (Takase et al., 2016) proposed an attention-based encoder-decoder model that adopts abstract

TABLE 3.1. Generated summaries via different MDS models. Different colors mean different thought groups.

Source Documents	a girl reported missing more than two years ago when she was 15 told police she escaped a home in illinois ... ... they recovered the child and arrested a 24-year-old man ... ... she was 15 when she disappeared. she escaped from the home in washington park earlier this week and went to police ...
HT	... she was also taken into custody. ...
FT	... the girl , who was 15 when she escaped from a home in washington park earlier this week. ...
ParsingSum-HT (Ours)	... a 24-year-old man were arrested and taken into custody. ...
ParsingSum-FT (Ours)	... she was 15 when she disappeared from the home. ...

meaning representation parser to capture structural syntactic and semantic information. The authors also pointed out that for natural language generation tasks in general, semantic information obtained from external parsers could help improve the performance of encoder-decoder based neural network model. Patrick et al. (Fernandes, Allamanis, and Brockschmidt, 2019) adopted named entities and entity coreferences for summarization problem. Jin et al. (Jin, Wang, and Wan, 2020c) enriched a graph encoder with semantic dependency graph to produce semantic-rich sentence representations. Song et al. (Song et al., 2020a) presented a LSTM-based model to generate sentences and the parse trees simultaneously by combining a sequential and a tree-based decoder for abstractive summarization generation.

Dependency parsing, an important linguistic knowledge that retains the intra-sentence syntactic relations between words, has been adopted and shown promising results in a variety natural language processing task (Deguchi, Tamura, and Ninomiya, 2019; Sun et al., 2019; Wang et al., 2020b; Cao et al., 2021; Wu et al., 2017). When it comes to document summarization, according to Hirao et al. (Hirao et al., 2004), no matter how the word order changes from the source documents to generated summaries, the dependency structures will keep consistent in most cases. Incorporating dependency structures into summarization models is crucial to retain the correct logics from source documents. The parsing information is usually formed as a tree structure that offers discriminate syntactic paths on arbitrary sentences for information propagation (Sun et al., 2019). The grammatical structure between the pair of words can be extracted from the dependency parser helping the model retain the syntactic structure. Therefore, in this thesis, we presented two multi-document summarization models by leveraging linguistic knowledge.

- **Model 1 ParsingSum.** The first work introduced a generic and flexible framework linguistic guided attention to incorporate dependency information into the Transformer based summarization models. We developed the proposed framework based on Flat Transformer (FT) and Hierarchical Transformer (HT), named ParsingSum-FT and ParsingSum-HT. Our proposed models can also be applied for both single and multiple document summarization. Table 3.1 is an example to illustrate why dependency information helps improve the quality of summaries. The data source is from Multi-News dataset (Fabbri et al., 2019a). The HT model can not distinguish who was arrested: it should be *"a 24-year-old man"* rather than *"she"*. In contrast, ParsingSum-HT (our model) shows consistent content with source documents. The potential reason is that the dependency parsing captures the relation between *"arrested"* and *"man"*, which keeps the token relations for summaries generation. We also find the FT model mingles two events within two sentences. However, the source documents show two events: (1) the disappearance of the girl in Illinois was at her age of 15; (2) she escaped from her Washington Park home two years later. Comparatively, ParsingSum-FT (our model) retains correct information. This is due to, from the linguistic perspective, a sentence is a linguistic unit that has complete meaning (Halliday et al., 2014). Furthermore, dependency parsing focuses on intra-sentence relations that help summaries retain correct syntactic structure. Figure 3.1 presented the framework of the proposed model ParsingSum. The proposed linguistic-guided attention mechanism is generic and flexible to be applied in different Transformer structures. Inside the model, the encoder is a representations learner to learn distinctive feature representations from the source documents and decoder is able to decipher representations into language domain for summary generation. More concretely, the document sets are first fed into a Transformer-based encoder for representation learning. Meanwhile, the source documents are passed into an external dependency parser to fetch the dependency relations. These relations and the Transformer's multi-head attention then be input into the linguistic-guided attention mechanism to construct the linguistic attention map. With the assistance of linguistic information, the model can grasp intra-sentence linguistic relations for summaries generation.

- **Model 2 DocLing:** Except lack of linguistic knowledge, Transfomer based summarization models face another chanllenge: only token-level positional encoding is not sufficient to capture document-level positional information. Missing document-level positional encoding significantly prevents models from detecting cross-document relationships. To solve the above-mentioned two

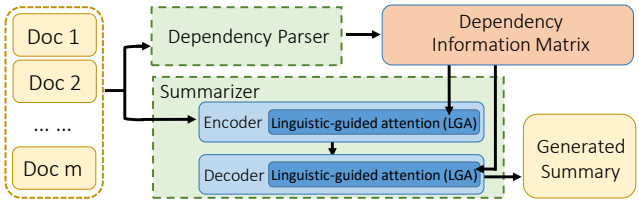


FIGURE 3.1. The framework of ParsingSum. Document sets are first fed into the encoder to generate the representations. In the meantime, these documents are input to a dependency parser to produce their sentence dependency information. The dependency information matrix will be further processed into a linguistic-guided attention mechanism and then fused with Transformer's multi-head attention, guiding the downstream summary generation.

problems, the second work proposed an encoding mechanism combining document aware positional encoding and linguistic-guided encoding for abstractive MDS. Figure 3.2 illustrates a general overview of the proposed method: DocLing. We constructed a document-aware positional encoding protocol to guide the encoding process and the selection of document-level positional encoding functions. Like most of the Transformer-based models, we added document-aware positional encoding with the input token embedding at the bottoms of the encoder stacks. Furthermore, we extended our efforts by encoding 45 distinct dependency relations into a dependency relation mask using a straightforward yet highly effective non-linear encoding strategy aimed at enhancing feature learning. The proposed linguistic-guided encoding method allows the model to better understand the relationship between each pair of words, and retains the correct dependency structure as well as grammatical associations when generating the summaries.

3.2 Methodology 1: ParsingSum

3.2.1 Dependency Information Matrix

Dependency grammar is a family of grammar formalisms that plays an important role in natural language processing. The dependency parser constructs several dependency trees that represent grammatical structure and the relations between *head* words and corresponding *dependent* words. To utilize these dependency information, we first adopted an external dependency parser (Dozat and Manning, 2017a), which can handle sentences of any length, to generate a set of dependency trees from multiple documents. The trees contain dependencies between any pair of dependent words

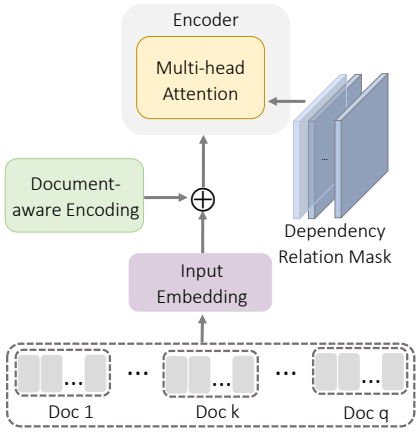


FIGURE 3.2. The framework of our proposed document-aware positional encoding and linguistic-guided encoding. Document-aware positional encoding serves as part of the input of the encoder; the proposed dependency relation mask will be incorporated with multi-head attention.

in one sentence. Let P denotes the dependency information matrix for one sentence. $p_{ij} \in P$ is a dependency weight between token t_i and token t_j. We simplified the definition of the weight as shown in Eq.(1):

$$p_{ij} = \begin{cases} 1 & t_i \ominus t_j \\ 0 & t_i \oslash t_j \end{cases} \tag{3.1}$$

where $t_i \ominus t_j$ indicates that t_i and t_j have a dependency relation, while $t_i \oslash t_j$ represents there is no dependency between the two tokens. To simplify the model, we consider the relations are undirected by ignoring the direction of *head* word and *dependent* word. For any pair of tokens, as long as there is a dependency between them, the dependency information matrix is assigned a value of 1, otherwise it will be set to 0. We hope to keep all dependency relations between the pair words in a simple yet effective manner.

3.2.2 Linguistic-Guided Attention Mechanism

In order to process source documents effectively and preserve salient source relations in the summaries, in ParsingSum, we presented a novel linguistic-guided attention mechanism to extend the Transformer architecture (Vaswani et al., 2017; Liu and Lapata, 2019a). Figure 3.3 depicts this mechanism on an exemplary sentence from Multi-News dataset (Fabbri et al., 2019a). linguistic-guided attention joins the dependency information matrix with the multi-head attention from source documents to generate syntactic-rich features. The linguistic-guided attention mechanism can be viewed as learning graph representations for the input sentences. Let $x_i^l \in \mathbb{R}^{d_{model}*1}$

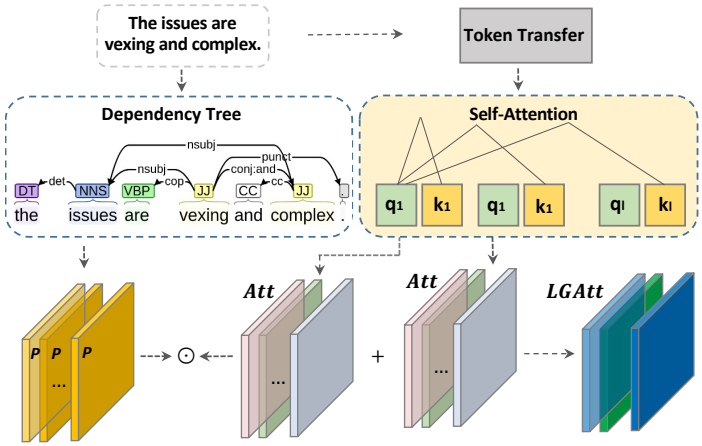


FIGURE 3.3. The linguistic-guided attention mechanism. The given exemplary sentence *The issues are vexing and complex.* is from Multi-News dataset (Fabbri et al., 2019a). Different properties of vocabularies and relations between words are included in the parsing information. The linguistic-guided attention mechanism incorporates the dependency information matrix P constructed from dependency trees of the input content and the Transformer's multi-head attention of this input content.

denotes the output vector of the last encoding layer of Transformer for token t_i. For the attention head $head_z \in Head(j = 1, 2, ..., h)$, h represents the number of head. We have:

$$\begin{aligned} q_{i,head_z} &= W^{q,head_z} x_i^l \\ k_{i,head_z} &= W^{k,head_z} x_i^l \\ v_{i,head_z} &= W^{v,head_z} x_i^l \end{aligned} \tag{3.2}$$

where $W^{q,head_z}$, $W^{k,head_z}$, $W^{v,head_z} \in \mathbb{R}^{d_k * d_{model}}$ are weight matrices. d_k is the dimension of the key, query and value. $q_{i,head_z}$, $k_{i,head_z}$, $v_{i,head_z} \in \mathbb{R}^{d_k * 1}$ are sub-query, sub-key and sub-values in different heads and we concatenate them respectively:

$$\begin{aligned} Q_i &= concat(q_{i,head_1}, q_{i,head_2}, ..., q_{i,head_h}) \\ K_i &= concat(k_{i,head_1}, q_{i,head_2}, ..., q_{i,head_h}) \\ V_i &= concat(v_{i,head_1}, q_{i,head_2}, ..., q_{i,head_h}) \end{aligned} \tag{3.3}$$

where $Q_i, K_i, V_i \in \mathbb{R}^{h * d_k * 1}$ are corresponding key, query and value for attention calculation. In ParsingSum, the linguistic-guided attention merges dependency information with multi-head attention in the following manner:

$$LGAtt_{ij} = \alpha M_{ij} \odot Att_{ij} + Att_{ij} \tag{3.4}$$

where

$$Att_{ij} = softmax\left(\frac{Q_i{}^T K_j}{\sqrt{d_k}}\right) \tag{3.5}$$

$$M_{ij} = Stack\_h(p_{ij}) \tag{3.6}$$

where α is a trade-off hyper-parameter to balance the linguistic-guided information M_{ij} and multi-head attention Att_{ij}. In order to fuse dependency weight p_{ij}, we build a function $stack\_h(\cdot)$ to repeat p_{ij} on the dimension of head to have the same size with $Att_{ij} \in \mathbb{R}^{h*1*1}$. $\odot$ denotes the element-wise Hadamard product. Then, we have:

$$Context_i = \sum_j LGAtt_{ij} \cdot V_j \tag{3.7}$$

where $Context_i$ represents the context vectors generated by linguistic-guide attention. Later on, two layer-normalization operations are applied to $Context_i$ to get the output vector of current encoder layer for token t_i:

$$x_i^{l+1} = LayerNorm\left(k_i + FFN\left(k_i\right)\right) \tag{3.8}$$

$$k_i = LayerNorm\left(x_i^l + Context_i\right) \tag{3.9}$$

where FFN is a two-layer feed-forward network with ReLU as activation function. Then, the learned feature representations are passed into multiple decoder layers that are fairly similar to the Flat Transformer structure (Gehrmann, Deng, and Rush, 2018).

3.3 Methodology 2: DocLing

In this work, we incorporated two types of encodings into Transformer-based abstractive MDS model: i) *document-aware positional encoding* considered document positional information; ii) *linguistic-guided encoding* incorporated dependency information into summarization process. The encodings will be introduced based on the following problem formulation and notations: given a set of N documents $D = \left(d^1, d^2, ..., d^N\right)$ on the same topic, the task of MDS is to generate a concise and informative summary Sum distilling knowledge from D. Let t_i^k denotes the $i\text{-}th$ token in the $k\text{-}th$ document $d^k (k = 1, 2, ..., N)$ in D. e_i^k represents the token embedding assigned to t_i^k by the Transformer model.

3.3.1 Document-aware Positional Encoding

For the token t_i^k from the source documents, the token positional encoding $Pos_{token_i}^k$ and document positional encoding $Pos_{doc_i}^k$ can be represented as:

$$\begin{aligned} Pos_{token_i}^k &= f_{token}\left(i\right) \\ Pos_{doc_i}^k &= f_{doc}\left(k\right) \end{aligned} \tag{3.10}$$

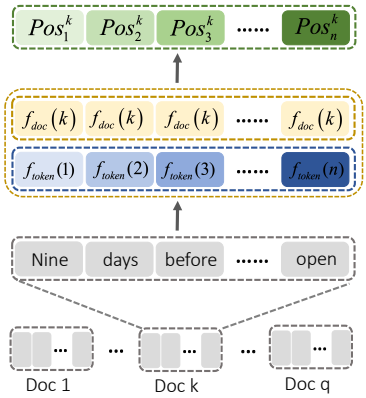


FIGURE 3.4. The proposed document-aware positional encoding. It contains a document-level positional encoding and a token-level positional encoding. The selection of document positional encoding functions is according to our proposed protocol.

where f_{token} and f_{doc} are encoding functions for token positional encoding and document positional encoding respectively. Different from the token positional embedding that considers the order of tokens, document positional embedding does not require the document order information as the order does not affect the MDS tasks. In order to find a proper f_{doc}, we designed a protocol with three considerations: (1) The encoding of each document should be unique. The purpose is to distinguish the documents and trace the source document for the tokens. (2) The values of encoding should be bonded. It will inevitably introduce large bias to certain documents if the encoding values are not bonded. (3) The values of encoding can not be remarkably larger than the value of token positional encoding. It will overwhelm the values of the token positional encoding if the document encoding values are too large, which impedes the model optimization process.

We adopted the $\sin$ function as document positional encoding function. Many other functions satisfying the document positional encoding protocol. We discussed their performances in Section 3.4.6. The final positional encoding Pos_i^k for token t_i^k combines the token-level and document-level positional encoding by a linear combination:

$$Pos_i^k = \alpha Pos_{doc_i}^{k'} + Pos_{token_i}^k \tag{3.11}$$

where

$$Pos_{doc_i}^{k'} = Stack\_dim_{token}(Pos_{doc_i}^k) \tag{3.12}$$

where $Stack\_dim_{token}(\cdot)$ is to repeat $Pos_{doc_i}^k$ for dim_{token} times to have the same

dimension with $Pos^k_{token_i}$. Then the overall input representations to the Transformer-based model are obtained by simply adding the token embedding and its corresponding positional encoding:

$$E^k_i = Pos^k_i + e^k_i \tag{3.13}$$

Figure 3.4 illustrates the process of proposed document-aware positional encoding. Given a set of documents (containing N documents), the document positional encoding combines with token positional encoding to form the document-aware positional encoding, which later serves as part of the input to the encoder of Transformer.

3.3.2 Linguistic-guided Encoding

We extended the dependency information matrix in ParsingSum by encoding 45 distinct dependency relations into a dependency relation mask using a straightforward yet highly effective non-linear encoding strategy aimed at enhancing feature learning. We constructed the three-order tensor Dep to place the dependency relations (the tokens discussed below are all from the same document, so the superscript k is omitted). The specific dependency relations $dep_{ij} \in Dep$ can be defined as below:

$$dep_{ij} = \begin{cases} v_{rel} & t_i \ominus t_j \\ 0 & t_i \oslash t_j \end{cases} \tag{3.14}$$

where $v_{rel} \in \mathbb{R}^{R*1}$ is the one-hot vector of dependency relations between token t_i and t_j. There are a variety of dependency relations between paired words in dependency parsing and R represents the total number of these dependencies. $t_i \ominus t_j$ indicates there is a dependency relation for t_i and t_j, while $t_i \oslash t_j$ represents no existing dependency between the two tokens. To encode these dependency relations into the Transformer-based models, we first transfered the dependency tensor into a dependency encoding weight through a two-layers encoding function:

$$m_{ij} = F_{depEnc}(dep_{ij}) \tag{3.15}$$

where F_{depEnc} contains two linear transformations and one LeakReLU non-linear mapping in between:

$$F_{depEnc}(x) = \text{Linear} \circ \text{LeakyReLU} \circ \text{Linear}(x) \tag{3.16}$$

where $\circ$ represents the concatenation of multiple sub-functions. In general, we discovered that the complexity of designing the encoding function for dependency information is crucial for model optimization. A too-naive encoding function may lack the ability to embed the information well enough; while an encoding function with

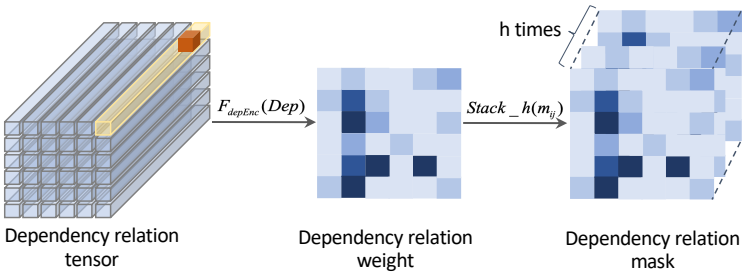


FIGURE 3.5. The transformation of dependency relation mask (right) from dependency relation tensor (left).

overly strong fitting abilities results in a slow training process and may cause failures in transforming the dependencies in an easy-optimizable manner.

Figure 3.5 shows the process of the transformation from dependency relation tensor Dep to dependency relation mask M_{ij}. Each fiber of the dependency relation tensor represents a one-hot vector for a specific dependency relation. Only the corresponding element of the one-hot vector has the value (highlight in red). The dependency relation weight m_{ij} is joined with the multi-head attention from source documents to generate syntactic-rich features in the following manner:

$$MHAtt(t_i, t_j, m_{ij}) = \sum_j \widetilde{A_{ij}} \cdot V_j \tag{3.17}$$

where

$$\widetilde{A_{ij}} = M_{ij} \odot A_{ij} + A_{ij} \tag{3.18}$$

$$A_{ij} = softmax\left(\frac{{Q_i}^T K_j}{\sqrt{dim}}\right) \tag{3.19}$$

$$M_{ij} = Stack\_h(m_{ij}) \tag{3.20}$$

where $Q_i, K_j, V_j \in \mathbb{R}^{h*d_k*1}$ are corresponding key, query, value for token t_i and t_j. dim is the dimension of the key, query and value. h is the number of attention heads. Both dim and h are fixed values that we followed the original settings in Transformer. In order to fuse dependency relation weight m_{ij} into dependency relation mask M_{ij}, function $Stack\_h(\cdot)$ is to repeat p_{ij} on the dimension of head to have the same size with $Att_{ij} \in \mathbb{R}^{h*1*1}$. $\odot$ denotes the element-wise Hadamard product. Then two layer-normalization operations are applied to get the output vector of the current encoder or decoder layer for the token t_i.

3.4 Experiments

3.4.1 Datasets

Multi-News Dataset (Fabbri et al., 2019a) is a large-scale English dataset containing various topics in news domain. It includes 56,216 document-summary pairs and it is further scattered with the ratio 8:1:1 for training, validation, and test respectively. Each document set contains 2 to 10 documents with a total length of 2103.49 words. The average length of the gold summaries is 263.66. Multi-XScience Dataset (Lu, Dong, and Charlin, 2020b) is a large-scale English dataset and it contains 40,528 document-summary pairs collected from scientific articles. The task of the Multi-XScience dataset is to generate the related work section of a target scientific paper based on the abstract of the same target paper and the abstracts of the articles it refers to. The dataset contains 30,369 training, 5,066 validation and 5,093 testing data. Samples have an average input length of 778 tokens and an average length of 116 tokens on the summary. WCEP-100 (Ghalandari et al., 2020b) consists of 10,200 document sets (8158 for training, 1020 for validation and 1022 for testing) with one corresponding human-written summary. The average length of the summaries are 32 words. For the ParsingSum model, our experiments were conducted using the Multi-News and WCEP-100 datasets. As for the DocLing model, we performed experiments on the Multi-News and Multi-XScience datasets.

3.4.2 Baselines

We compared our proposed method with the following strong baselines: *LexRank* (Erkan and Radev, 2004b) computes textual unit salience based on the eigenvector centrality algorithm using heuristic features in the similarity graph-based sentence representations. *TextRank* (Mihalcea and Tarau, 2004) leverages the graph-based ranking formula, deciding on the importance of a text unit representative within a graph built for information extraction. *SummPip* (Zhao et al., 2020) constructs sentence graphs by incorporating both linguistic knowledge and deep neural representations. *Maximal Marginal Relevance (MMR)* (Carbonell and Goldstein, 1998b) combines query relevance and information novelty from source documents, benefiting summarization in reducing redundancy while remaining the most salient information. *Bidirectional recurrent neural network (BRNN)* superimposes two RNNs of opposing directions on the same output according to RNN states. *Transformer* (Vaswani et al., 2017) follows an encoder-decoder structure based on attention mechanism, which has been extensively utilized in a wide range of natural language processing

TABLE 3.2. Models comparison on Multi-News test set. We reran all the compared models under the same environment. The best results for each column are in bold.

Models	ROUGE-1	ROUGE-2	ROUGE-L
LexRank	37.92	13.10	16.86
TextRank	39.02	14.54	18.33
MMR	42.12	13.19	18.41
SummPip	42.29	13.29	18.54
BRNN	38.36	13.55	19.33
FT	42.98	14.48	20.06
Hi-MAP	42.98	14.85	20.36
HT	36.09	12.64	20.10
ParsingSum-HT (Ours)	37.34	13.00	20.42
ParsingSum-FT (Ours)	**44.32**	**15.35**	**20.72**

tasks[1]. *CopyTransformer* restricts abstractive summarizer to copy tokens from source documents. *Pointer-Generator (PG)* (See, Liu, and Manning, 2017b) equips with the coverage mechanism between the pointer network and the standard sequence-to-sequence attention model. *Hierarchical MMR-Attention Pointer-generator (Hi-MAP)* model (Fabbri et al., 2019a) integrates sentence representatives with hidden-state-based MMR into a standard pointer-generator network, an end-to-end model for abstract summarization. *Hierarchical Transformer (HT)* (Liu and Lapata, 2019a) captures relationships across multiple paragraphs via the hierarchical Transformer encoders and flat Transformer decoders[2].

3.4.3 Automatic Evaluation Metrics

We evaluated the models by using ROUGE scores (Lin, 2004a) and BERTScore (Zhang et al., 2020c). Unigram and bigram overlap (ROUGE-1 and ROUGE-2 scores) are adopted to indicate the literal quality of generated summaries. ROUGE-SU score is a unigram-based co-occurrence statistic, bringing out the soft skip bigram by computing both the skip-bigram and unigram. ROUGE-L adopts the longest common subsequence algorithm to count the longest matching vocabularies. ROUGE F1 scores are considered in our work[3]. BERTScore is an automatic language evaluation metric for text generation based on contextual token embeddings of the pre-trained BERT (Devlin et al., 2019a).

[1]We implemented the Transformer model based on https://github.com/Alex-Fabbri/Multi-News/tree/master/code/OpenNMT-py-baselines

[2]We trained the HT model on one GPU for 100,000 steps with batch-size 13,000.

[3]The scores are computed with ROUGE-1.5.5 script with option "-c 95 -2 -1 -U -r 1000 -n 4 -w 1.2 -a -m"

TABLE 3.3. Models comparison on WCEP-100 test set. The best results for each column are in bold.

Models	ROUGE-1	ROUGE-2	ROUGE-L
HT	23.20	5.78	17.45
FT	23.41	6.64	17.93
ParsingSum-HT (Ours)	24.03	6.42	18.31
ParsingSum-FT (Ours)	**26.45**	**7.06**	**18.98**

3.4.4 Experimental Settings

ParsingSum: We equipped the proposed linguistic-guided attention on both Hierarchical Transformer (HT) and Flat Transformer (FT) architectures. Two models are thus derived: ParsingSum-HT and ParsingSum-FT. For ParsingSum-HT, we followed the implementation of the HT model by using six local Transformer layers and two global Transformer layers with eight heads[4]. For ParsingSum-FT, we followed FT model settings and adopt four encoder layers and four decoder layers[5]. For training, we used *Adam* optimizer ($\beta 1$=0.9 and $\beta 2$=0.998). The dropout rates of both encoder and decoder are set to 0.1. The initial learning rate is set to 1×10^{-3}. The first 8000 steps are trained for warming up and the models are trained with a multi-step learning rate reduction strategy. Deep Biaffine dependency parsing (Dozat and Manning, 2017a) are used to generate dependency information for these source documents.

DocLing: To have a fair comparison, we kept all the experimental settings consistent throughout all experiments. In our Transformer-based model, eight encoder layers and decoder layers are adopted. The Biaffine parser (Dozat and Manning, 2017b) is used for generating dependency relations among the source documents. Our model adopts 45 dependency relations. We used *Adam* optimizer ($\beta 1$=0.9 and $\beta 2$=0.998) for model parameter optimization. The initial learning rate of the model is set to 1×10^{-3} and 0.1 dropout rate is set for both the encoder and decoder. The trade-off hyper-parameter α is set to 0.1. In the training phase, the first 8×10^3 steps are trained for warming up and the models are trained with a multi-step learning rate reduction strategy. In the experiments, the model accumulates gradients and updates once every four iterations. The minimum and maximum lengths of the generated summaries are set to 200 and 300 words for the Multi-News dataset, while 110 and 300 words for the Multi-XScience dataset.

TABLE 3.4. The analysis of fusion weights of linguistic-guided attention on Multi-News validation set. The best results for each column are in bold.

Models	ROUGE-1	ROUGE-2	ROUGE-L
HT	36.02	12.57	20.05
ParsingSum-HT (α=1)	36.71	12.79	20.27
ParsingSum-HT (α=2)	35.64	12.18	19.80
ParsingSum-HT (α=3)	**36.74**	**12.86**	**20.29**
FT	42.81	14.25	19.81
ParsingSum-FT (α=1)	43.69	14.67	19.95
ParsingSum-FT (α=2)	**43.84**	**15.01**	**20.50**
ParsingSum-FT (α=3)	43.61	14.92	20.13

3.4.5 Model Performance of ParsingSum

Overall Performance

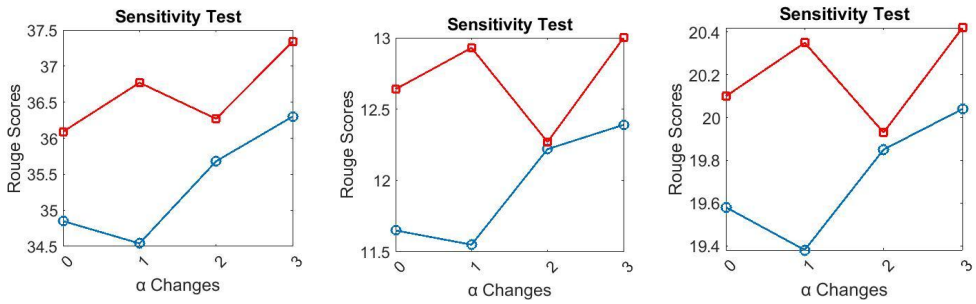
We evaluated the proposed ParsingSum-HT, ParsingSum-FT and compare them with multiple mainstream models on both Multi-News and WCEP-100 datasets. For fair comparisons, we reran all the compared models under the same environment. For Multi-News dataset, as shown in Table 3.2, the ParsingSum-HT model receives higher ROUGE scores (across all ROUGE-1, ROUGE-2 and ROUGE-L) steadily compared to the original HT model. The linguistic-guided attention helps the model raise 1.25 on ROUGE-1 score, 0.36 on ROUGE-2 score, and 0.32 on ROUGE-L respectively. It indicates the outstanding capability of ParsingSum models to retain the intention of original documents when generating summaries. A similar phenomenon shows on the ParsingSum-FT model. More specifically, ParsingSum-FT surpasses FT model 1.34 on ROUGE-1 score, 0.87 on ROUGE-2 score, and 0.66 on ROUGE-L score, which shows the effectiveness of linguistic-guided attention on the Transformer-based models. It is worth noting that the proposed ParsingSum-FT is able to outperform its baseline (i.e., FT model) by a large margin and also receives the highest ROUGE scores across all the compared methods. The effect of linguistic-guided attention can be verified on the WCEP-100 dataset. The ROUGE results can be improved on both two version of Transformer based summarization models. These results indicate the outstanding capability of linguistic-guided attention to retain the intention of original documents when generating summaries.

[4]We trained the HT model on one GPU for 100,000 steps with batch-size 13,000.

[5]We implemented the FT model based on https://github.com/Alex-Fabbri/Multi-News/tree/master/code/OpenNMT-py-baselines. We trained the FT model for 20,000 steps with batch-size 4096 on one GPU.

TABLE 3.5. Human evaluation results on the Multi-News dataset. The best results for each column are in bold.

Models	Fluency	Informativeness	Consistency
Hi-MAP	2.53	2.80	2.33
FT	2.47	2.67	2.60
HT	2.20	2.13	2.40
ParsingSum-HT	2.73	**2.93**	**2.87**
ParsingSum-FT	**2.87**	2.87	2.73



(ROUGE-1) (ROUGE-2) (ROUGE-L)

FIGURE 3.6. The performance of ParsingSum-HT on small (in blue) and large batch-size setting (in red).

Human Evaluation

Although ROUGE are the standard evaluation metrics for summarization tasks, they focus on lexical matching instead of semantic matching. Therefore, in addition to the automatic evaluation, we accessed model performance by human evaluation in a semantic way. We invite three annotators who research natural language processing to evaluate the performance of five models (Hi-MAP, FT, HT, ParsingSum-FT, ParsingSum-HT) independently. For each model, 30 summaries are randomly selected from the Multi-News dataset. Three criteria are taken into account to evaluate the quality of generated summaries: (1) Informativeness: how much important information does the generated summary contain from the input document? (2) Fluency: how coherent are the generated summaries? (3) Consistency: how closely the information in the generated summaries are consistent with the input documents? Annotators are asked to give scores from 1 (worst) to 5 (best). Table 4.5 summarizes the comparison results of five summarization models. For each model, the score of each criterion is computed by averaging the score of all summary samples. The results demonstrate that the Transformer based models equipped with linguistic-guided attention are able able to generate higher quality summaries than the baseline models in terms of informativeness, fluency, and consistency. These human evaluation results further validate the effectiveness of our proposed linguistic-guided attention mechanism.

TABLE 3.6. Performance of ParsingSum-HT via different fusion methods on Multi-New validation set. The best results for each column are in bold.

Models	ROUGE-1	ROUGE-2	ROUGE-L
ParsingSum-HT (P0.25)	19.50	3.40	12.59
ParsingSum-HT (G0.25)	16.84	1.92	11.36
ParsingSum-HT (G8)	20.18	3.55	13.00
ParsingSum-HT (α=3)	**36.74**	**12.86**	**20.29**

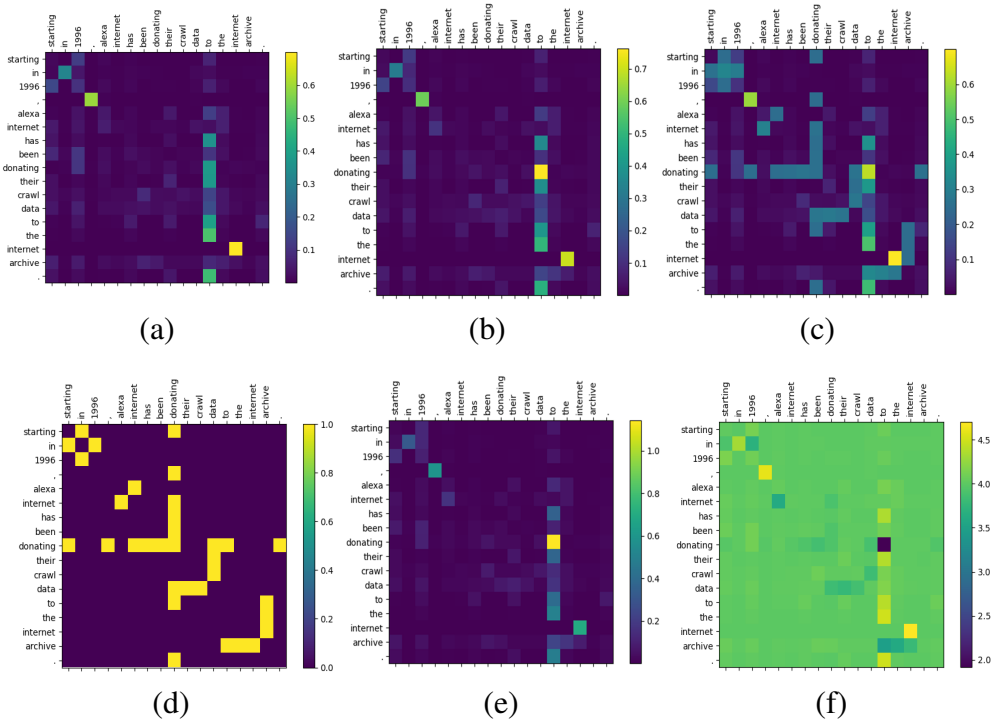


FIGURE 3.7. Visualization of different fusion methods. (a) HT model; (b) ParsingSum-HT (α=1); (c) ParsingSum-HT (P0.25); (d) dependency parsing matrix; (e) ParsingSum-HT (α=3); (f) ParsingSum-HT (G0.25).

Analysis

We further analyzed the effects of the trade-off parameter α and batch-size in ParsingSum. We also examined and discuss different manners to incorporate parsing information into the proposed model.

The Analysis of the Fusion Weights. The trade-off factor α controls the intensity of attention from a linguistic perspective to be fused with multi-head attention. To analyze its importance, we conducted experiments by setting α to 0, 1, 2, and 3 ($\alpha = 0$ denotes the naive Transformer model without linguistic-guided attention) on the two proposed models on the validation set. The results are shown in Table 3.4. Generally, there is an increasing trend with the increment of α. This rising trend further proves

assigning a relatively larger α in a suitable range can improve the performance of summarization models.

The Analysis of Batch-size. Batch-size is considered to have a great effect on the mini-batch stochastic gradient descent process of model training (Smith et al., 2018) and it will thus further affect the model performance. To validate it empirically, we trained the model with small/large batch-size (the small batch-size is 4,500 and the large one is 13,000) of the ParsingSum-HT model. The experiments are conducted with different α. The results in Figure 3.6 show that smaller batch-size reduces the performance on all the evaluation metrics. Interestingly, the ROUGE scores of the small batch-size setting are steadily increasing with α changes from 1 to 3; when the model is trained with large batch-size, the increasing trend is retained but the ROUGE scores are jittering when α equals two. It indicates different batch-sizes have different sensitivities towards the change of α.

The Analysis of the Fusion Methods. How to integrate the parsing information into the Transformer-based model is important in our work. In addition to the fusion method introduced in Section 3.2.2, we attempted several other fusion methods under a small batch-size setting of the ParsingSum-HT model: (1) Direct fusion. Weight the dependency parsing matrix and add it directly to the multi-head attention. It denotes as ParsingSum-HT (P0.25):

$$LGAtt_{ij} = 0.25M_{ij} + Att_{ij} \tag{3.21}$$

(2) Gaussian-based fusion. We adopted the idea from (Li et al., 2020c) and apply Gaussian weights to the product of the dependency information and the multi-head attention. The Gaussian weights are set to 0.25 (ParsingSum-HT (G0.25)) and 8 (ParsingSum-HT (G8)):

$$LGAtt_{ij} = \frac{(1 - M_{ij}Att_{ij})^2}{0.25} + Att_{ij} \tag{3.22}$$

$$LGAtt_{ij} = \frac{(1 - M_{ij}Att_{ij})^2}{8} + Att_{ij} \tag{3.23}$$

Figure 3.7(a) and 3.7(d) represent the heatmap of the HT model and dependency parsing matrix. Figure 3.7(b), 3.7(c), 3.7(e), and 3.7(f) illustrate the attention maps of different fusion methods. Table 3.6 presents the performance of the mentioned fusion methods on Multi-New validation set. ParsingSum-HT with α=3 receives the best results for all ROUGE scores. The potential reason is that through direct fusion and Gaussian fusion, the scale of the original multi-head attention has been overwhelmed, leading to posing the dependency information in a dominant position. In this case, the normal gradient backpropagation process has been disturbed. The experiment

TABLE 3.7. Performance comparison on the Multi-News dataset. We reran all the baseline models under the same settings. “CopyTrans" represents CopyTransformer. The best results for each column are in bold.

Models	ROUGE-1	ROUGE-2	ROUGE-SU	BERTScore
LexRank	37.92	13.10	12.51	0.83
TextRank	39.02	14.54	13.08	0.83
SummPip	42.29	13.29	16.16	0.84
MMR	42.12	13.19	15.63	0.84
BRNN	38.36	13.55	14.65	0.83
Transformer	25.82	5.84	6.91	0.80
CopyTrans	42.98	14.48	16.91	0.84
PG	34.13	11.01	11.58	0.83
Hi-MAP	42.98	14.85	16.93	0.83
HT	36.09	12.64	12.55	0.84
DocLing	**44.35**	**15.04**	**17.97**	**0.85**

TABLE 3.8. Performance comparison on the Multi-XScience dataset. We reran all the baseline models under the same settings. “CopyTrans" represents CopyTransformer. The best results for each column are in bold.

Models	ROUGE-1	ROUGE-2	ROUGE-SU	BERTScore
LexRank	**31.31**	5.85	9.13	0.83
TextRank	31.15	5.71	9.07	**0.84**
SummPip	29.66	5.54	8.11	0.82
MMR	30.04	4.46	8.15	0.83
BRNN	27.95	5.78	8.43	0.83
Transformer	28.34	4.99	8.21	0.82
CopyTrans	26.92	4.92	7.50	0.83
PG	30.30	5.02	9.04	**0.84**
Hi-MAP	30.41	5.85	9.13	0.81
HT	25.31	4.23	6.64	0.83
DocLing	30.93	**6.06**	**9.57**	**0.84**

results indicate that a direct summation of the weighted dependency parsing matrix and multi-head attention may damage the original attention. On the other hand, a “soft” fusion (when α is adopted) of these two attentions can achieve promising results.

3.4.6 Model Performance of DocLing

Overall Performance

In this section, we compared our proposed model with several strong baselines and list the comparison results in Table 3.7 (Multi-News) and Table II (Multi-XScience). The results of our proposed model on the Multi-News dataset show the best overall results on both ROUGE scores and BERTScore. To give a fair comparison, we reran all the baseline models. It is observed that our model performs particularly

TABLE 3.9. Ablation study of our model on Multi-News and Multi-XScience dataset. “doc-pos en" and “depen en" stand for document-aware positional encoding and linguistic-guided encoding.

Dataset	Model Variants	ROUGE-1	ROUGE-2	ROUGE-SU
Multi-News	w/o doc-pos en	44.16	15.06	17.74
	w/o depen en	43.73	14.86	17.37
	Full Model	44.35	15.04	17.97
Multi-XScience	w/o doc-pos en	28.81	5.53	8.56
	w/o depen en	29.69	5.62	8.86
	Full Models	30.93	6.06	9.57

well on R-SU than other models. It gains 1.06 improvement to the second best, *Hi-MAP*. Given that R-SU takes more skip-bigram plus unigram-based co-occurrence statistics into account, it contains additional comprehensive information to evaluate the models. The BERTScore on different models shows relative marginal differences. However, our proposed model still achieves the best among all the evaluate models, which indicates our proposed model can generate high-quality summaries in a semantic level. We also evaluated our proposed models based on the Multi-XScience datasets. Comparing the *Transformer* baseline models and our model with document-aware positional encoding and linguistic guided encoding, we observed that these two encodings help to improve the performance by 2.59 on R-1, 1.07 on R-2 and 1.36 on R-SU. The results on the Multi-XScience dataset show that our model performs better than most of the models. Our proposed model does not achieve the best results on all evaluation metrics because the proposed model is based on the Transformer models which are dataset sensitive. This means the Transformer-basd models do not always work well on all the MDS datasets. This phenomenon can also be found in the paper (Zhao et al., 2020; Pasunuru et al., 2021b) and (Jin and Wan, 2020). In these paper, the Transformer-based model (*CopyTransformer*) shows poor results on DUC-2004 dataset[6] although it works well on the Multi-News dataset. A potential reason is Multi-XScience and DUC-2004 datasets have higher novel n-grams score than Multi-News dataset (Fabbri et al., 2019a; Lu, Dong, and Charlin, 2020b). For example, paper (Lu, Dong, and Charlin, 2020b) reported that the proportion novel of unigrams/bigrams/trigrams/4-grams in the gold summaries of the Multi-News dataset is 17.76/57.10/75.71/82.30, which are much lower than that of Multi-XScience dataset (42.33/81.75/94.57/97.62). The Transformer models may not work very well on datasets with higher novel n-gram scores.

[6]http://duc.nist.gov

Ablation Study

To better understand the contribution of document-aware positional encoding and linguistic-guided encoding techniques to overall model performance individually, we conducted an ablation study on the proposed model on both Multi-News and Multi-XScience datasets. Table 3.9 presents the results. The experiments confirm that the proposed two encoding methods perform considerably better than the model without them. This is due to (1) document-aware positional encoding has the capability of capturing cross-document information in MDS; (2) with linguistic-guided encoding, dependency relations within the source documents are well preserved, enabling the summarization model to effectively learn a much more faithful syntactic structure than that working on the model without it.

Encoding Strategies

In addition to the model performance evaluation, we reported our findings on different encoding functions and the ways to incorporate the encoding.

(1) Document-aware Positional Encoding Strategies. We evaluated the contribution of different document positional encoding functions. All these functions satisfy the proposed protocol described in Section 3.3.1. The experience results are shown in the upper part of Table 3.10. $\sin$ function helps the MDS model achieve the best ROUGE score and the combination of $\sin$ and $\cos$ produce similar results. However, $\cos$ function greatly reduces the model performance. The reason could be related to the document number in a document set of Multi-News dataset. Most of the document sets contain two documents in the Multi-News dataset. When applying $\cos$ on two documents, the value differences for the two encodings is smaller than what the $\sin$ function provides, which means $\cos$ has less distinguishing ability than $\sin$. This may result in lower model performance for MDS tasks. Additionally, we also tried to adjust α in Equation (2). Results are shown in the lower part of Table 3.10. We tested the model performance on validation set when $\alpha = 0.1, 0.5, 1$ and observe model perform best when $\alpha = 0.1$. Therefore, we fixed this hyper-parameter to 0.1 and report the final results on the test set.

(2) Document-aware Positional Encoding Protocol. To verify the proposed three considerations of document encoding functions, we selected some other functions except sin and cos, and the functions are not satisfy the conditions proposed in 3.3.1 for experiments. We also randomly assigned values to document positional encoding to verify the effectiveness of our chosen function. The results are shown in Table 3.11. We observed that the performance are not well when (1) the document positional encoding of each document is the same (SameEncoding); (2) the values of

TABLE 3.10. Performance of our model using different document positional encoding strategies. The strategies include different encoding functions (upper) and different document positional encoding weights(lower). iter(A, B) means to use functions A and B alternately. Values obtained from the validation set based on the Multi-News dataset.

Models	ROUGE-1	ROUGE-2	ROUGE-SU
$\sin(x)$	43.80	14.74	17.59
$\cos(x)$	42.82	14.49	16.71
$iter(\sin(x), \cos(x))$	43.56	14.43	17.38
$iter(\sin(0.1x), \cos(0.1x))$	43.66	14.52	17. 47
α= 0.1	44.11	14.81	17.74
α= 0.5	43.68	14.54	17.45
α= 1	43.80	14.74	17.59

TABLE 3.11. Performance of models with functions that do not meet the document positional encoding protocol. Values obtained from the validation set based on the Multi-News dataset.

Models	ROUGE-1	ROUGE-2	ROUGE-SU
SameEncoding	42.82	14.28	16.63
$y = x$	42.25	14.08	16.25
$y = 2x$	42.57	14.06	16.60
$y = 5x$	40.56	12.06	15.13
$y = 10x$	38.94	11.50	14.24
Random	43.19	14.67	16.87

document positional encoding are not bonded ($y = x$, $y = 2x$, $y = 5x$, $y = 10x$); and (3) the values of document positional encoding are remarkable larger than the values of token positional encoding ($y = 10x$); (4) randomly assign values to document positional encoding (Random).

(3) Linguistic-guided Encoding Strategies. There are 45 dependency relations existing in the Biaffine parser. Some dependency relations have a great influence on the generated summaries; and vice versa. This section discussed how to encode these various relations into multi-head attention mechanism by considering their importance. The performance of different linguistic-guided encoding methods is shown

TABLE 3.12. Performance of our model based on different linguistic-guided encoding methods. Values obtained from the validation set based on the Multi-News dataset.

Models	ROUGE-1	ROUGE-2	ROUGE-SU
Arithmetic sequence	43.71	14.54	17.43
Arithmetic sequence (core)	43.79	14.57	17.47
Arithmetic sequence (root)	43.89	14.64	17.55
One-hot (one layer)	43.15	14.40	17.03
One-hot (F_{depEnc})	44.11	14.81	17.74
Added on values	42.41	13.89	16.47

TABLE 3.13. Human evaluation on the Multi-News. The best results for each column are in bold. “CopyTrans" represents CopyTransformer.

Models	Fluency	Informativeness	Conciseness
Transformer	2.50	1.97	2.50
CopyTrans	2.60	2.60	2.83
Hi-MAP	3.07	2.87	2.97
DocLing	**3.13**	**3.10**	**3.20**

in Table 3.12. The importance of dependency relations in the first three methods are manually set and the following two are automatically learned. “Arithmetic sequence" represents a sequence with the values of $1, R-1/R, R-2/R..., 1/R$, which means the dependency relations at the top of the list have a larger weight. R denotes the number of dependency relations in total. The sequence of dependency relation list in the first methods is constructed according to the sequence of the occurrence of dependency relations in the source documents. We selected the top-8 dependents from the official core dependents of clausal predicates[7] to build the relation lists for “Arithmetic sequence (core)". “Arithmetic sequence (root)" is to assign the largest weight to the root word since the dependency relation “root" is proven to be the most important token in the syntax dependency tree (Wang et al., 2020b). “One-hot (one layer)" means the one-hot representation of dependencies with only one linear transformation between the dependency relation tensor and the dependency relation mask. The “One-hot (one layer)" model performs substantially poorer than the one-hot encoding model with non-linear function F_{depEnc}. It is because non-linearity enlarges the learning capability of encoding functions significantly. The“One-hot (F_{depEnc})" represents our final model. The F_{depEnc} function can outperform all arithmetic sequence models since it delegates the construction of dependency relations to a non-linear learner. It enables the model to learn the importance of gradient descent directly. From another point of view, besides the addition of the linguistic-guided encoding on keys and queries within self-attention, we also tried to add the encoding on values. However, model performance dropped greatly. We hypothesized the reason is that keys and queries are adopted to calculate attention, but values are the final receptors of attention. Small changes in values will have a large influence on the model optimization process.

Human Evaluation

Apart from automatic evaluation, we conducted a human evaluation to assess the quality of the generated summaries on three aspects: **text fluency** checks whether

[7] https://universaldependencies.org/docs/en/dep/

TABLE 3.14. Generated summaries of different models given the same source documents. “CopyTrans" represents CopyTransformer. Different colors represent different thought groups.

Source Docs	a nine-year-old boy from los banos has completed quite the journey on tuesday , as he swam through the san francisco bay , all the way to alcatraz island and back . james savage is hoping to be the youngest swimmer on record to make the swim . the title is currently held by a 10-year-old . it ' s been quite an emotion day in san francisco , but it ended with smiles and cheers as james walked on shore , after hours in the bay . james began the swim at 7 : 00 a.m. , and it took him a little over two hours to complete the feat . he struggled for a big out in the open water , about 30 minutes into the swim . james said the waves were hitting him , and the current was too strong , making it difficult for him to fight them . in the end , however , with help from his coach and a promise from his father , james found his second wind , and was off . " i wanted to turn back and they kept yelling , ' i believe i you you can do it ' , " said james . " we kind of pushed each other . finally got him going . smooth sailing all the way .....james savage , a 9-year-old boy from los banos , calif.....
Transformer	james savage has been swimming off the san francisco bay area since he was 14 years old , but he ' s not the only one who ' s been in the water for more than a century . " i ' ve never seen anything like this before , " he tells nbc bay area . " it ' s a great thing to do . i ' m not going to be able to do it . " he says he ' ll be swimming in the bay of san francisco ' s bay in san francisco when he ' d swim to the beach , and he ' s ⟨blank⟩⟨blank⟩⟨blank⟩.....
CopyTrans	james savage , a 9-year-old boy who completed the tiring trek last year , was able to set a record as the youngest swimmer to make the journey to the island and back on june 14 , reports nbc bay area ..... james tells ksee . " he ' s not going to do anything about it , " he says , adding that his second wind was off the rails . " it ' s really hard to get to the bottom of the road , " savage says of his father . " it was just a slap on the wrist . it was a slap in the water . it ' s ⟨blank⟩⟨blank⟩⟨blank⟩.....
Hi-Map	james savage is hoping to be the youngest swimmer on record to make the swim to the former prison , reports the san francisco chronicle . " i wanted to turn back and they kept yelling , ' i believe i you you can do it ' , " james told ksee . " we kind of pushed each other . finally got him going . smooth sailing all the way , " he says . " i ⟨blank⟩⟨blank⟩⟨blank⟩.....
Ours	james savage , a 9-year-old boy from los banos , calif. , has completed the swim at 7 : ⟨unk⟩and 57 pounds , was able to set a record as the youngest swimmer to make the journey to the alcatraz island and back on june 14 , nbc news reports. " i wanted to turn back and they kept yelling , ' i believe i you can do it , ' " james tells ksee . " we kind of pushed each other . finally got him going . smooth sailing all the way , " .....

the summary is natural, well-formed, and both syntactically and semantically correct; **conciseness** assesses whether the summary is concise and without repeated or useless information; **informativeness** examines whether the summary keeps the salient information from the source documents. We randomly sampled 10 examples from the Multi-News dataset (Fabbri et al., 2019a). Three experienced researcher are invited to score summaries (from 4 models) on the above aspects. The score range is 1-5 (1 means very bad; 5 means very good). The final scores for each model are averaged across different examples and raters. The results are listed in Table 4.5. The text fluency score of our model is 3.13, which is higher than 2.50 of *Transformer*, 2.60 of *CopyTransformer*, and 3.07 of *Hi-Map*, which means the summaries generated by our model are more natural and well-formed. In terms of the score of informativeness, our model achieves 3.10 and is higher than the second-best model (*Hi-Map*) by 0.23, indicating our model is better at capturing the most important information from different sources. Moreover, the generated summaries by our model are more concise and better at reducing redundant information, which could be concluded by the conciseness score.

Case Study

Table 3.14 presents the generated summaries from four models: *Transformer*, *CopyTransformer*, *Hi-Map*, and our models. In this example, the *Transformer* model only captures “james savage has been swimming off the san francisco bay area" (in red) but takes the age wrong. It should be 9 in fact. Besides, *Transformer* model also generates something that are not supported in the source document (in orange). For the *CopyTransformer*, the salient information (in green) is in the generated summary. However, this model also outputs unsupported text (in orange). The *Hi-Map* model misses some key information (e.g. the red highlight in the source document). In contrast, the summary generated by our proposed model keeps the significant information and shows content consistent with the source documents. It could demonstrate that our model equipped with the proposed informative encoding mechanism could generate summaries more accurately than the other comparing models.

3.5 Conclusion for the Chapter

In this chapter, we presented two methods to incorporate linguistic-guided encoding for abstractive multi-document summarization. In the first work, the proposed linguistic guided attention mechanism can be seamlessly incorporated into multiple mainstream Transformer-based summarization models and can outperform existing Transformer-based methods by a large margin. We developed two models

based on Flat Transformer (FT) and Hierarchical Transformer (HT). The proposed ParsingSum-HT and ParsingSum-FT incorporate dependency relations with Transformer's multi-head attention for summaries generation. Based on this work, we encoded 45 distinct dependency relations into a dependency relation mask. We also proposed an effective and informative encoding mechanism to encode the multi-document positional information and give a general protocol to guide the selection of document encoding functions. We conducted extensive experiments on two benchmark datasets and the results demonstrate the superior performance of the proposed two encoding methods. The analysis of various settings of the document-aware positional encoding and linguistic-guided encoding can help researchers understand the intuitiveness of the proposed model and could serve as an informative reference to the MDS research community.

Chapter 4

Disentangling Specificity for Abstractive Multi-document Summarization

4.1 Introduction

In the preceding chapter, we delved into addressing the challenges of Multi-Document Summarization (MDS) by examining linguistic and document positional perspectives, unraveling the interconnections among various documents. In this chapter, our focus extends beyond merely exploring connections; we also delve into the distinctions between input documents. Our objective is to generate summaries that are both informative and comprehensive, synthesizing key insights from multiple input documents. Some researchers tried to establish the connections not only at word-level relations but also sentence, paragraph and document levels. They employ hierarchical Transformer structures (Liu and Lapata, 2019a; Li et al., 2020b; Jin, Wang, and Wan, 2020a; Song, Chen, and Shuai, 2022) to forge connections among documents. The high-level Transformer encodes the paragraph representations from different documents. Besides, some existing works incorporated graph information (Fan et al., 2019; Li et al., 2020b; Pasunuru et al., 2021b; Wang et al., 2020a) to build connections among documents. However, these methods are not specifically designed for extracting specific features and therefore they ignore the specific information contained in each document in a document set.

Nonetheless, the extraction of specific information is crucial with the following reasons: (1) In a collection of documents, each document contains not only the common information but also has specific contents that distinguish it from other documents. These specific information contain unique facts, viewpoints, and details (Fabbri et al., 2019a). Extracting these specific details enhance the comprehensiveness of the resulting summary. Additionally, some essential information may be exclusive to a particular document, yet it plays a pivotal role in obtaining a comprehensive grasp

of the entire document set. Therefore, a high-quality MDS summary should not only be able to capture document commonality but can also comprehensively consider the specific information from each document, covering various dimensions to meet the user's demand for a comprehensive understanding of the documents (Wolhandler et al., 2022). (2) Focusing on the extraction of specific information helps reduce redundancy, rendering the summary more concise and informative. Clustering-based MDS methods (Goldstein et al., 2000; Wan and Yang, 2008; Nayeem, Fuad, and Chali, 2018; Pasunuru et al., 2021b; Ernst et al., 2022) can be used to group similar sentences or pieces of information and remove redundancy. After removing the redundant information, the remaining information in each document can be viewed as implicitly specific information. However, the specificity within these remaining information cannot be explicitly guaranteed to be distinctive between documents.

In order to address this issue, our intuition is not only to capture the overall information in a document set but also to distinguish the specificity of each document and learn representations of document specificity which will be considered in the summary generation process. To this end, we proposed DisentangleSum — a simple yet effective summarization model that disentangles document uniqueness with a set of document-specific representation learners. In order to optimize the learning of specific representations, we further proposed an orthogonal constraint to encourage the specific representations obtained from a pair of documents to be distinctive from each other. Based on the constraint, we designed an objective function that can transform the quadratic increment of the losses between each of the paired documents into linear to cope with a large number of documents in a set. We summarize our contributions as follows:

- We presented DisentangleSum, an innovative MDS model that is capable of disentangling specific information from each document in a set, leading to more comprehensive summary generation. To the best of our knowledge, we are the first to consider the specific information for deep learning based MDS task.

- To incentivize the document-specific learner to retain document specificity information, we proposed an orthogonal constraint. This constraint encourages the document-specific representation vectors to align vertically with each other, ensuring a semantic separation between them.

- Experimental results on two MDS datasets demonstrate the effectiveness of DisentangleSum. We additionally offered comprehensive analyses from multiple perspectives to investigate the underlying mechanisms of DisentangleSum and circumstances of the proposed model can work.

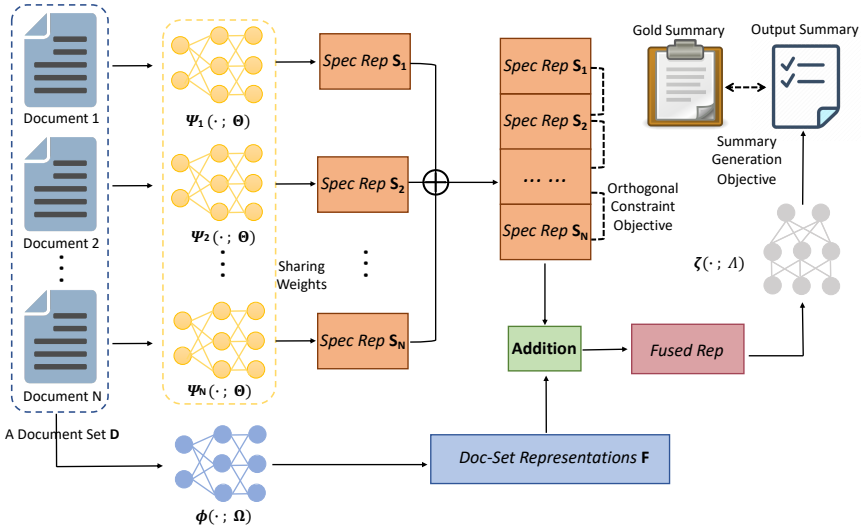


FIGURE 4.1. The overall framework of the proposed Disentangle-Summodel. One to N documents are processed individually by a set of specific encoders $\{\psi_1, \psi_2, \psi_3, ..., \psi_N\}$ with sharing weights into specific representations $\{\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3, ..., \mathbf{S}_N\}$. These specific representations are then concatenated into one specific vector and finally added to the document-set representation generated by another document-set encoder ϕ for summary generation. During the optimization process, an orthogonal constraint is introduced to encourage the learned specific features that are dissimilar to each other.

4.2 Our Approach

In this section, we provided an overview of the proposed model, DisentangleSum, by describing how to incorporate document disentangling specificity representation learning into the summarization framework. We introduced the orthogonal constraint applied during the training of document-specific features. In Figure 4.1, individual documents in a set are processed by specific Transformer-based encoders to learn document-specific features. Simultaneously, in order to capture overall document-set context, these documents are also concatenated into a document and fed into a Transformer based document-set encoder. The overall document-set features are then added to the document-specific features for decoding.

4.2.1 Problem Formulation

In the context of MDS tasks, each document set can have a varying number of documents. For illustration purposes, let's consider a document set $\mathbf{D} = (\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, ..., \mathbf{d}_N)$ consisting of N input documents related to a specific topic or sharing common information. In our approach, we utilized the specific encoder $\psi_i(\cdot; \Theta)$ for document

$\mathbf{d}^i$, where Θ represents the learnable parameters. These specific encoders generate specific representations S_i for each document, and collectively, they form the specific representations $\mathbf{S}$ for the entire document set. Additionally, we employed a document-set encoder $\phi(\cdot;\Omega)$ with learnable parameters Ω to obtain document-set representations $\mathbf{F}$. The target is to generate a concise summary output $\mathbf{O}$ that synthesizes all important contents from input documents by considering both specific representations $\mathbf{S}$ and document-set representations $\mathbf{F}$.

4.2.2 Document Specific Representation Learner

In a document set, the specific representation learner aims to identify the specific information within each document. To achieve this, we introduced a specific encoder to encode document $\mathbf{d}^i$ in the same document set $\mathbf{D}$:

$$\mathbf{S}_i = \psi_i(\mathbf{d}^i;\Theta), \tag{4.1}$$

Under the setting of MDS, the number of input documents can vary within a document set (e.g., Multi-News dataset have two to ten documents per set). To address this variability, we presented a design where the learnable parameters, denoted as Θ, are shared across a set of N specific encoders instead of assigning a separate specific encoder to each document in the set. The rationale behind this approach stems from the fact that documents with identical indexes in different document sets are unrelated in terms of their contents. Consequently, maintaining multiple separate specific encoders for each indexed document is not reasonable. Subsequently, we concatenated these specific representations to obtain the overall specific features of a document set:

$$\mathbf{S} = \mathbf{S}_1 \oplus \mathbf{S}_2 \oplus \mathbf{S}_3 \oplus ... \oplus \mathbf{S}_N, \tag{4.2}$$

$\oplus$ is the concatenation operation in this work. To enable sufficient expressive power for representations to be decoded, we also obtained the document-set representations $\mathbf{F}$ out of a document set D by:

$$\mathbf{F} = \phi(D;\Omega), \tag{4.3}$$

Next, we combined the document-set representations and specific representations by performing an element-wise addition and decoding them into summarization outputs:

$$\mathbf{O} = \zeta(\alpha \cdot \mathbf{F} + \mathbf{S};\Lambda), \tag{4.4}$$

Here, α serves as a trade-off factor to control the weight balancing between the document-set representations and specific representations. The decoder function

$\zeta(\cdot;\Lambda)$, parameterized by Λ, is responsible for decoding the intermediate features into concise summaries.

4.2.3 Orthogonal Constraint within the Training of Document Specific Features

To guide the learning of specific features, we imposed an orthogonal constraint between pairs of specific features $\mathbf{S}_i$ and $\mathbf{S}_j$. The document-specific loss, which promotes dissimilarity between specific features, is defined as:

$$L_{spec} = \sum_{i} \sum_{j} \left\| \mathbf{S}_i{}^{\top} \mathbf{S}_j \right\|_F^2, \tag{4.5}$$

where $\|\cdot\|_F^2$ is the squared Frobenius norm. To encourage dissimilarity between specific features, we aimed for a smaller inner product between each pair of specific feature vectors, promoting orthogonality. This ensures that the specific features of each document within the same set are as distinct from each other as possible. As the specific encoder learns, it captures the unique essence of each individual document, thereby retaining specific content. However, when a document set contains more than two documents, the computation of specific representation objectives between every pair of documents grows exponentially. To address this, we introduced a circle-paired loss objective function, which effectively transforms the exponential growth into linear growth, suppressing computational complexity. Formally, we have:

$$L_{spec} = \sum_{i=1}^{N} L_{spec}^{i} , \tag{4.6}$$

$$L_{spec}^{i} = \begin{cases} \left\| \mathbf{S}_i{}^{\top} \mathbf{S}_{i+1} \right\|_F^2 & i \neq N \\ \left\| \mathbf{S}_N^{\top} \mathbf{S}_1 \right\|_F^2 & i = N \end{cases}, \tag{4.7}$$

The objective function calculates the specific feature costs between each document and the subsequent document in the set, with the last document computed against the first one.

4.2.4 Overall Objectives

The proposed framework aims to train a high-quality summarization model that incorporates specific representations from each document. This is achieved through two key components: an orthogonal constraint for distinct document representations and a supervised cross-entropy loss concerning gold summaries:

$$L_{total} = L_{gen} + \beta \cdot L_{spec}, \tag{4.8}$$

$$L_{gen} = -\sum_{k=1}^{M} \eta(\hat{\mathbf{O}}_k, \mathbf{O}_k) \log(p(\mathbf{O}_k)), \tag{4.9}$$

where β is a balance factor, $p(\mathbf{O}_k)$ is one shard[1] of the predictive summary from the DisentangleSum model. $\hat{\mathbf{O}}_k$ denotes corresponding true labels. M represents the number of shard within the generated summary. The calculation of $\eta(\cdot, \cdot)$ in a summarization task is different from other tasks such as text classification. $\eta(\cdot, \cdot)$ indicates the evaluation function between prediction and ground truth, the widely used ROUGE evaluation are adopted here.

4.3 Experiments

4.3.1 Datasets & Evaluation Metrics & Baselines

We assessed the effectiveness of the proposed method on Multi-News(Fabbri et al., 2019a) and Multi-XScience (Lu, Dong, and Charlin, 2020c) datasets which satisfy that documents contain the specific information in a document set. Both datasets are truncated to 500 tokens. We used standard summarization evaluation metrics ROUGE[2] (Lin, 2004b) which are based on word-matching and BERTScore[3] (Zhang et al., 2020b) which is based on semantically matching. Specifically, we used ROUGE-1 (R-1), ROUGE-2 (R-2), ROUGE-SU (R-SU) scores. Additionally, we employed the coverage rate (Grusky, Naaman, and Artzi, 2018) to quantify the amount of information retained in the generated summaries compared to input documents. This provides insights into the effectiveness of the proposed disentangling specificity representations in preserving important information. We invited three Ph.D students in the NLP area to examine the performance of four different models. The raters are asked to rate each summary along three dimensions: comprehensive, coherence, and relevance. 50 randomly sampled source documents from the Multi-News dataset. The score range from one to five (one means very bad; five means very good). The final scores are averaged across different cases and raters.

We compared with the following strong baselines: *LexRank* (Erkan and Radev, 2004b), *TextRank* (Mihalcea and Tarau, 2004), *MMR* (Carbonell and Goldstein, 1998b), *BRNN*, *Vanilla Transformer (VanillaTrans)* (Vaswani et al., 2017) and its variant *CopyTransformer (CopyTrans)*, *Pointer-Generator* (See, Liu, and Manning, 2017b),

[1] Following the implementation in https://github.com/Alex-Fabbri/Multi-News/blob/master/code/OpenNMT-py-baselines/onmt/utils/loss.py, shards are segments when computing losses.

[2] The parameters of ROUGE are -c 95 -2 -1 -U -r 1000 -n 4 -w 1.2 -a -m.

[3] The model type of BERTScore is bert-base-uncased.

TABLE 4.1. Performance comparison on the Multi-News dataset.

Models	R-1	R-2	R-SU	BS
LexRank	37.92	13.1	12.51	0.83
TextRank	39.02	14.54	13.08	0.83
MMR	42.12	13.19	15.63	0.84
BRNN	38.36	13.55	14.65	0.83
VanillaTrans	25.82	5.84	6.91	0.8
CopyTrans	42.98	14.48	16.91	0.84
PG	34.13	11.01	11.58	0.83
Hi-MAP	42.98	14.85	16.93	0.83
HierTrans	36.09	12.64	12.55	0.84
SummPip	42.29	13.29	16.16	0.84
SAGCopy	43.98	15.21	17.65	-
HeterGraphSum	43.62	14.99	17.29	**0.85**
HiTrans	44.62	15.57	18.06	-
DocLing	44.35	15.04	17.97	**0.85**
DisentangleSum	**45.95**	**16.32**	**19.23**	**0.85**

Hi-MAP (Fabbri et al., 2019a), *Hierarchical Transformer (HierTrans)* (Liu and Lapata, 2019a), *SummPip* (Zhao et al., 2020), *SAGCopy* (Xu et al., 2020c), *HeterGraphSum* (Wang et al., 2020a), *Highlight-Transformer (HiTrans)*(Liu et al., 2021), *DocLing* (Ma et al., 2022). The models with the best performance are bolded for each column in the following Tables.

4.3.2 Implementation Details

During model training, the initial learning rate is set to 2. The training strategy involves a warm-up phase for the first 8,000 steps, followed by multi-step learning rate reduction. The batch size is set to 4,096, and the models are trained for 20,000 steps using the *Adam* optimizer. Both the encoder and decoder consist of four transformer layers, and positional encoding is applied. The dropout rate is 0.2. The trade-off factor for specific features (α) is set to 0.01, and the trade-off factor for specific loss (β) is set to 0.001. The word embedding size for source documents is set to 512 dimensions. We conducted all the experiments on one NVIDIA 3090 GPU with one Intel i9-10900X CPU upon Ubuntu 22.04.3 LTS Operation System. For the minimum and maximum lengths of generated summaries, Multi-News has 200 and 300 words, while Multi-XScience has 110 and 300 words.

4.3.3 Main Results

This section is designated for validating the model's effectiveness from four perspectives: (1) verifying the comprehensiveness of the generated summaries; (2) evaluating the overall performance through automated evaluations; (3) assessing the model's

TABLE 4.2. Performance comparison on the Multi-XScience dataset.

Models	R-1	R-2	R-SU	BS
LexRank	31.31	5.85	9.13	0.83
TextRank	31.15	5.71	9.07	**0.84**
MMR	30.04	4.46	8.15	0.83
BRNN	27.95	5.78	8.43	0.83
VanillaTrans	28.34	4.99	8.21	0.82
CopyTrans	26.92	4.92	7.50	0.83
PG	30.30	5.02	9.04	**0.84**
Hi-MAP	30.41	5.85	9.13	0.81
HierTrans	25.31	4.23	6.64	0.83
SummPip	29.66	5.54	8.11	0.82
DocLing	30.93	**6.06**	9.57	0.84
DisentangleSum	**31.81**	5.90	**9.88**	**0.84**

TABLE 4.3. Human evaluation results on the Multi-News dataset. The final scores are averaged across different cases and raters. "Compr", "Coher" and "Relev" indicate comprehensiveness, coherence and relevance.

Models	Compr	Coher	Relev
Vanilla Trans	2.28	2.13	2.46
CopyTrans	2.70	2.27	2.78
DocLing	3.10	2.78	2.93
DisentangleSum	**3.87**	**3.21**	**3.13**

performance through human evaluations, assessing the effectiveness of extracting specific information, and ensuring the comprehensive, coherence and relevance of the generated summaries; (4) giving significance analysis.

Coverage Score

We conducted a comparison based on coverage score (Table 4.4) between DisentangleSum and three Transformer-based models, namely Vanilla Transformer, Copy-Transformer, and DocLing. These models share a similar structure but do not consider document specificity. Coverage score measures the percentage of words in the generated summary that are part of an extractive fragment with the input documents. The higher coverage score indicates that its corresponding model can produce summaries with richer information in the source documents. From the results, it is observable that DisentangleSum achieves the highest coverage score on both two datasets and outperforms its counterparts by a large margin, indicating the proposed DisentangleSum model can generate more comprehensive summaries that preserve more information from the original documents.

TABLE 4.4. Coverage score comparison on Multi-News and Multi-XScience datasets.

Models	Multi-News	Multi-XScience
Vanilla Trans	19.76	18.98
CopyTrans	46.78	15.82
DocLing	49.62	20.10
DisentangleSum	**52.99**	**22.68**

Overall Performance

Table 4.1 and Table 4.2[4] shows the proposed DisentangleSum model receives outstanding performance in most of the cases. On the Multi-News dataset, DisentangleSum outperforms the second-best model, attaining 1.6 improvement on ROUGE-1, 1.28 improvement on ROUGE-2, and 1.26 improvement on ROUGE-SU. Particularly, the ROUGE-SU score received 7% improvement over the second-best model. Similarly results are shown on Multi-XScience data as well. Compare with the second-best model for each column, DisentangleSum raises the ROUGE-1 score from 30.93 to 31.81 and the ROUGE-SU score from 9.57 to 9.88. The superior results can be consistently gained because the proposed DisentangleSum model has been empowered with the ability to precisely grasp the document set and the document-specific features for a better summary generation.

Human Evaluation

We conducted human evaluations to assess summary quality in terms of Specificity (Speci), Comprehensiveness (Compr), Coherence (Coher), and Relevance (Relev), aiming to detect diverse viewpoints from multiple documents. Comprehensive refers to the extent to which it covers the essential information present in the source documents. Coherence measures how well the content in a summary is logically connected and flows smoothly. Relevance refers to the degree to which the information presented in the summary is pertinent and directly related to the source documents. We invited three Ph.D students in the NLP area to examine the performance of four different models. 50 randomly sampled source documents from the Multi-News dataset. The score range from one to five (one means very bad; five means very good). The final scores (shown in Table 4.5) are averaged across different cases and

[4]The code for model SAGCopy (Xu et al., 2020c) and HiTrans (Liu et al., 2021) is not publicly available and they did not provide results on the Multi-XScience dataset. Additionally, since MultiX-Science does not contain labels for extractive summarization, hindering the the HeterGraphSum (Wang et al., 2020a) from being implemented on it.

Models	Speci	Compr	Coher	Relev
VT	1.67	2.28	2.13	2.46
CT	2.33	2.70	2.27	2.78
DocLing	2.89	3.10	2.78	2.93
DisentangleSum	**3.16**	**3.87**	**3.21**	**3.13**

TABLE 4.5. Human evaluation results on the Multi-News dataset.

raters. The results consistently favor DisentangleSum across all four human evaluation metrics. An example from the MultiNews (Fabbri et al., 2019a) dataset is shown in Table 4.6. The document set talks about how to discourage smoking, and each document discusses this topic from different angles. Doc #1 indicates the ugliest colour serves an important purpose: discouraging smoking, Doc #2 lists the statistics related to smoking, and Doc #5 discusses smoking from a legal point of view. Existing works provide summaries that miss some specific information from the source documents. For example, the summary generated by the DocLing (Ma et al., 2022) model fails to include the statistical information presented in Doc #2, resulting in the omission of important specific details from the source documents. These indicated the summaries generated by our model can cover more specific information from the source documents, and exhibit better coherence and relevance.

Significance Analysis

To assess the statistical significance of our model's results, we performed one-tailed paired t-test using the evaluation metrics ROUGE from the Multi-News dataset. While BERTScore is an effective measure of summary quality, the distinctions between various models were relatively minor, making it unsuitable for t-test application. Our findings, present in Table 4.7, revealed that all p-values were less than the significant threshold 5e-2. This outcome suggests that the DisentangleSum exhibits statistical significance, indicating that the observed differences in evaluation metrics are unlikely to be random and are more likely attributable to the structure of the model.

4.4 Model Analyses

4.4.1 Objective Function Selections

In MDS, each document in a set shares common content while also having unique information. We examined the effectiveness of models by incorporating shared features and specific representations through different objective functions. This evaluation aims to shed light on the importance of capturing both common and specific

TABLE 4.6. Source documents and generated summaries. The document set discusses the same event, but each document has specific content that sets it apart. Texts in the same color indicate the same thought groups.

Document Set	**Doc #1**: the world's ugliest color has been described as "death," "dirty" and "tar," but this odious hue is serving an important purpose: discouraging smoking. pantone 448 c, a "drab, dark brown" also called "opaque couché," was specifically selected after three months and multiple studies by research agency gfk. the agency was hired by the australian government to find a color that was so repugnant that if it was on tobacco products, it would dissuade people from smoking ... **Doc #2**: ...3.260 billion in december 2015.3 know your limits – changes to australia ' s duty free tobacco allowance smoking prevalence rates abs national aboriginal and torres strait islander social survey, 2014-15 the proportion of aboriginal and torres strait islander people aged 15 years and over who were daily smokers was 38.9 % in 2014-15 , down from 44.6 % in 2008 and 48.6 % in 2002 ... . **Doc #5**: ... in may, previously passed legislation will go into effect requiring all packs of cigarettes to be standardized. tobaccos products will be stripped of brightly colored branding and replaced with a sludge-like color . . but does the stripped-down , " ugly " packaging really reduce smoking ...
Vanilla Trans (Vaswani et al., 2017)	–australia's news agency says it's time to get rid of certain types of ⟨unk⟩products. the australian government has approved a ban on <unk> products , which include ...
CopyTrans[5]	-the world's ugliest color will be helpful in smoking rates in their country, according to a team of experts ... researchers found that pantone publications, including pantone 448c visually, are chock full of " ugly " reactions ...
DocLing (Ma et al., 2022)	... world's ugliest color - will be stripped of colored branding and replaced with a sludge-like color .in may, previously passed legislation will go into effect requiring all packs of cigarettes to be standardized ...
DisentangleSum (Ours)	–the world's ugliest color is serving an important purpose: ... more likely to deter smoking from reaching for their next pack of cigarettes ... it will go into effect requiring to be standardized ... found that smokers aged 15 years and over half a year ... in smoking rates than those in 2002 and 48.6 % in 2002 ...

Models	LexRank	TextRank	MMR	BRNN	VT	CT	PG
p-value	2.65e-2	4.53e-2	1.70e-3	3.56e-2	2.01e-2	9.23e-3	2.46e-2
Models	Hi-MAP	HierTrans	SummPip	SAGCopy	HGS	HiTrans	DocLing
p-value	1.77e-2	3.17e-2	1.95e-3	1.23e-2	1.17e-2	1.2e-2	3.16e-3

TABLE 4.7. ROUGE score p-value from one-tailed paired t-test on Multi-News dataset.

TABLE 4.8. Models performance with different objective functions on Multi-News validation dataset. "SSL", "TL", "SL", "CPL", "DPL" denote shared specific loss, triple loss, shared loss, circle-paired-loss and dense-paired-loss. "R" and "N" indicates randomly sort documents in the same document set and normalization.

Objectives	R-1	R-2	R-SU
SSL	44.64	15.47	17.83
TL	44.15	14.98	17.52
SL	44.10	15.00	17.47
CPL	45.16	15.39	18.48
CPL-R	45.03	15.09	18.40
DPL	44.74	15.33	18.05
DPL-N	44.41	14.93	17.86

information in MDS and understand the impact of different training objectives on model performance.

Specific-Shared Loss V.S. Triplet Loss

Inspired by the good performance for specific features, we intuitively thought that disentangling the shared feature may further improve the model performance. Here shared feature refers to common information shared by multiple documents in one set. To obtain the shared features, we incorporated a shared encoder $\omega(\cdot;\nu)$ to learn the shared representations $\mathbf{H_i}$ from the document $\mathbf{d_i}$ by:

$$\mathbf{H_i} = \omega(\mathbf{d_i};\nu), \tag{4.10}$$

where ν is a set of learnable parameters. We expected that, in one set, the shared features of each document should be similar and the specific features are distinguishable from each other. To achieve so, we attempted two objective functions to encourage the document-shared features to be similar and specific features to be distinctive from each other.

(1) Specific-shared loss (SSL) represents by:

$$L_{total} = L_{gen} + \beta \cdot L_{spec} + \gamma \cdot L_{shared}\ , \tag{4.11}$$

$$L_{shared} = \sum_{i=1}^{N} L^{i}_{shared} \ , \tag{4.12}$$

$$L^{i}_{shared} = \begin{cases} \|\mathbf{H}_i - \mathbf{H}_{i+1}\|_p & i \neq N \\ \|\mathbf{H}_N - \ \mathbf{H}_1\|_p & i = N \end{cases} , \tag{4.13}$$

$$L_{spec} = \sum_{i=1}^{N} \left\| \mathbf{S}_i^{\top} \ \mathbf{H}_i \right\|_F^2 , \tag{4.14}$$

where γ is a balance factor. For one document d_i, we expected its specific representations to be orthogonal with its shared representations.

(2) Triplet loss (TL) is inspired by contrastive representation learning from positive and negative samples (Schroff, Kalenichenko, and Philbin, 2015). For one document d_i, its shared feature $\mathbf{H_i}$ and specific feature $\mathbf{S_i}$ can be seen as an anchor and a negative sample, respectively. The positive sample can be another shared feature map $\mathbf{H_j}$ from document d_j. Note that d_i and d_j are in the same document set. The final objective function can be:

$$L_{total} = L_{gen} + \beta \cdot L_{triplet} \ , \tag{4.15}$$

$$L_{triplet} = \sum_{i=1, j \neq i}^{N} max(\|\mathbf{H}_i - \ \mathbf{H}_j\|_2^2 + \|\mathbf{H}_i - \ \mathbf{S}_i\|_2^2) \ , \tag{4.16}$$

Table 4.8 reveals that the performance of the SSL-equipped model surpasses that of the TL-equipped model. However, it falls short when compared to the results achieved by the Disentangle model's objective. We hypothesized the conflict between shared features and the document-set features causes this phenomenon during the optimization of summary generation. As a result, we decided not to incorporate the shared and specific representations together in the proposed objective function.

Specific Loss V.S. Shared Loss

Given the better performance of specific-shared loss than triplet loss, we further examined the roles played by specific features and the shared features separately. We compared the results of specific loss and shared loss. The total objective function to generate shared features can be defined as:

$$L_{total} = L_{gen} + \gamma \cdot L_{shared} \ , \tag{4.17}$$

where the calculation of L_{shared} is equal to Equation (4.12). Table 4.8 shows the performance on the Multi-News validation dataset. The model equipped with shared loss obtains lower performance than that equipped with specific loss.

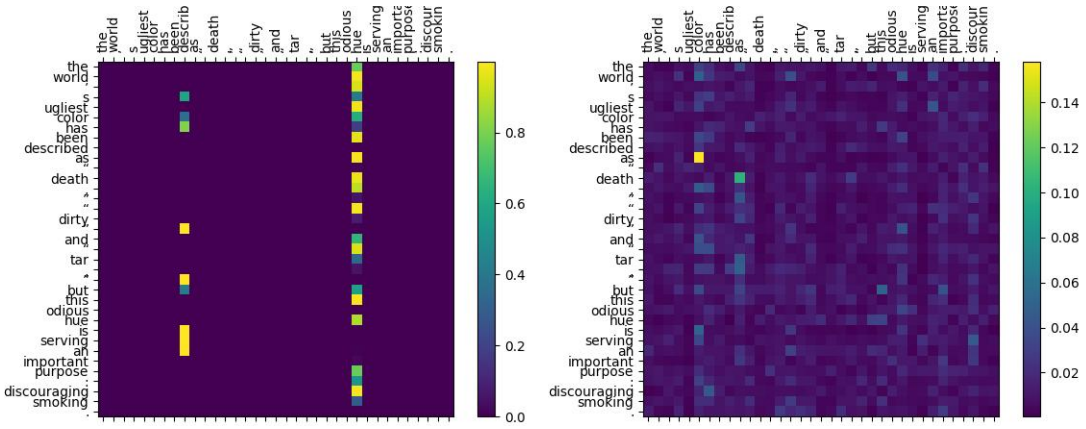


FIGURE 4.2. Attention maps of learned specific features and shared features. Sentence “the world’s ugliest color has been described as "death," "dirty" and "tar," but this odious hue is serving an important purpose: discouraging smoking.” is from the first document of Table 4.6.

Furthermore, to dig out why the specific loss has a comparative advantage in summaries generation, we visualized the attention maps (Figure 4.2) of specific features and shared features from the last encoding layer. Interestingly, the attention-specific encoder is mainly focused on the individual words “hue" which is the specific information for #1 document. However, the heatmap of shared features is more scattered than the specific features. This may because specific features concentrate on important information of each document while shared features do not. Consequently, we opt not to select the objective function associated with shared representations for our main experiment.

The Selection of Specific Loss

This section investigate two design options of specific loss: circle-paired loss (CPL), introduced in Section 4.2.3, and dense-paired loss (DPL). DPL indicates the specific feature loss will be computed from each pair of documents in the same document set. We conducted two experiments in this subsection:

(1) Compare the overall performance of DisentangleSum equipped with CPL and DPL. We evaluated the models on Multi-News dataset, analyzing their performance with different objectives. The results in Table 4.8 indicate that CPL outperforms DPL across all three evaluation metrics.

(2) Explore the impact of document number on specific loss. To investigate the relationship between specific loss and the number of documents, we divided the

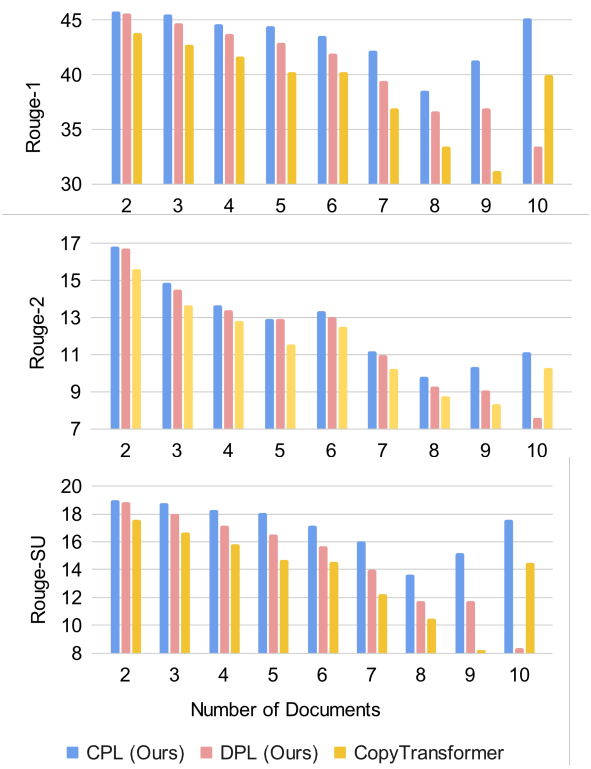


FIGURE 4.3. ROUGE scores of DisentangleSum with circle-paired-loss (CPL), DisentangleSum with dense-paired-loss (DPL), and CopyTransformer on document sets containing two to ten documents.

Multi-News validation set into subsets based on the document set size. We compared the model performance trained with CPL and DPL on these subsets. Figure 4.3 illustrates that DisentangleSum with CPL outperforms DPL and CopyTransformer across all subsets in terms of three ROUGE scores. Notably, when the document set size is two, the results of DPL and CPL are quite similar for the ROUGE-1 score. However, as the number of documents increases, the model trained with DPL experiences a significant performance drop, while the model trained with CPL exhibits a slower decline. This trend holds for ROUGE-2 and ROUGE-SU scores as well. Besides, from the perspective of computational complexity, as the number of documents increases, the document pairs in DPL increase quadratically (e.g. 10 documents yield 45 pairs), while CPL does not.

In order to exclude the impacts of the order of documents in a document set and the loss scale, we further conducted two experiments: (1) Based on CPL, we randomly sorted documents in the same document set; (2) Based on DPL, we adjusted the loss scale through normalization, dividing the right head side of Equation 4.5 by N^2. The results (in Table 4.8) ruled out the interference of these two factors. The performance differences may be that for MDS tasks, the documents in

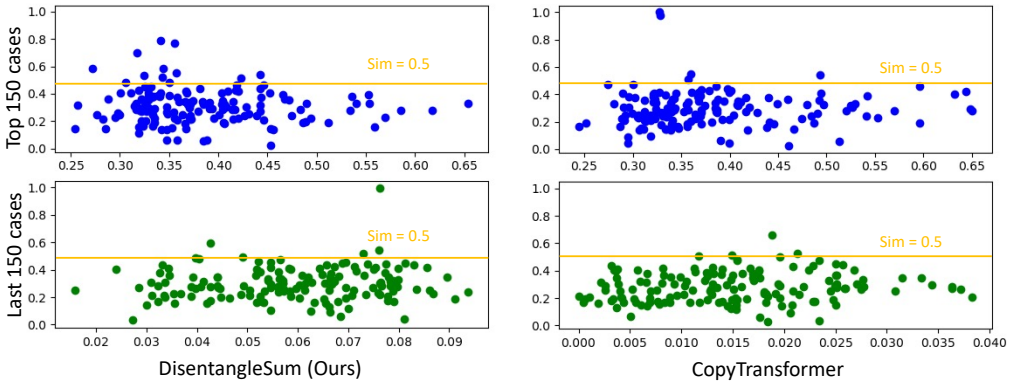


FIGURE 4.4. The distribution of document similarity scores in the Top 150 and Last 150 cases. The X-axis and Y-axis of each sub-figure are ROUGE-SU scores (scale to $0 \sim 1$) and documents similarity scores, respectively. The orange line represents the document similarity score equal to 0.5.

the same document set describe topic-relevant concepts, yet with some document-specific information. The constraint may be too strong by imposing a model to learn document-specific representations completely different between documents, which in turn may incur a confused model and less "informative" representations learned.

4.4.2 DisentangleSum Performances with Different Inter-Document Similarities

The purpose of this subsection is to examine the relationship between DisentangleSum performance and inter-document similarities. We defined a simple function to calculate the document similarity within a document set using statistical analysis:

$$Sim(D) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{2 \cdot overlap(d^i, d^j)}{\mathbf{N}(\mathbf{N}-1)}, \tag{4.18}$$

where $\mathbf{N}$ represents the document number in each document set. It calculates the content overlap between each pair of documents within the document set. We evaluated the DisentangleSum and CopyTransformer models by calculating ROUGE scores for document sets and ranking them. We analyzed the Top 150 and Last 150 cases, finding average similarity scores of 0.308 and 0.299 for DisentangleSum, and 0.293 and 0.281 for CopyTransformer. Figure 4.4 shows: (1) DisentangleSum's Top 150 cases have slightly higher document similarity scores than the Last 150 cases. (2) DisentangleSum's Top 150 cases have more instances with similarity scores above 0.5 compared to Last 150 cases and CopyTransformer's Top 150 cases.

TABLE 4.9. Model performance on Multi-News validation set by tuning specific feature trade-off factor α and loss trade-off factor β.

Variants	R-1	R-2	R-SU	Variants	R-1	R-2	R-SU
α = 1	43.60	13.94	17.30	β = 1	44.15	15.17	17.72
α = 0.1	43.36	13.67	17.26	β = 0.1	44.24	15.16	17.77
α = 0.01	45.16	15.39	18.48	β = 0.01	44.64	15.45	18.09
α = 0.001	44.67	15.50	17.90	β = 0.001	45.16	15.39	18.48
w/o Spec Feat	43.66	14.79	17.39	w/o Spec Loss	43.66	14.79	17.39

These findings suggest that the proposed model tends to perform better when the document similarity score is higher. The potential reason is in one document set when the overlap between each document is relatively large, the ratio of the uniqueness of each document is relatively small. Models that do not explicitly capture document-specific information may struggle to capture the specific details from the source documents. The DisentangleSum model, designing to retain document-specific information, performs better in such cases when the document similarity score is higher.

4.4.3 Hyperparameter Scale of Models

We perform a hyperparameter study to examine the effectiveness of specific feature trade-off factor α and loss trade-off factor β, controlling the trade-off strength of fetching the document specific information and document-set information. The results are shown in Table 4.9. Both the weights of α and β are controlled by searching the grid [1, 0.1, 0.01, 0.001, 0]. The experiments of evaluation α is performed under β equals to 0.001; while the examination β is conducted by setting α to 0.01. By setting either specific feature weights or specific loss weights to 0s, the model performance is significantly degraded. It suggests the positive contribution of grasping documentary unique information. Interestingly, with the increasing of specific feature trade-off factor α from 0.001 to 0.01, the ROUGE scores generally have an increasing trend. But the score goes up and then goes down when α is from 0.01 to 1. The optimal choice of the hyper-parameter α falls in the middle of the evaluated values, which is 0.01. Similar results show for the experiments of loss trade-off factor β. Generally, 0.001 is recommended for β to achieve the best performance. The experimental results indicate that the existence of the document-specific representation learner and the orthogonal constraint of document-specific feature generation is important. Meanwhile, setting large α and β obstructs model optimization and summary generation.

4.5 Conclusion for the Chapter

In this chapter, we introduced DisentangleSum, a framework to disentangle document-specificity for better abstractive MDS representations. To optimize the specific feature learning, we applied an orthogonal constraint to encourage the document-specific learner to catch document-specific information. The experiments on two prevalent datasets show the superior performances of the proposed model over other counterparts. Furthermore, we also provided extensive analyses that reveal DisentangleSum exhibits broader coverage of input documents and better preservation of document-related information. These analyses help researchers understand the intuitiveness of the proposed model and could serve as an informative reference to the MDS research community.

Chapter 5

Exploring Transformer-based Multi-document Summarization: An Empirical Investigation

5.1 Introduction

The innovation and contemporary developments of Transformer architecture (Vaswani et al., 2017) thrives multi-document summarization (MDS) (Ma et al., 2020). Notably, the methodologies introduced in the preceding two chapters are rooted in the Transformer framework. This impetus prompts an exploration into the intricacies and behaviors of established Transformer-based MDS models. Through the comprehensive analyses, we aimed to provide a thorough understanding of MDS and its intricacies within the MDS model framework. We undertook a comprehensive investigation from five distinct perspectives covering the Transformer-based MDS model design pipeline:

- Document input perspective: we conducted experiments to quantitatively assess the impact of document separators from a standpoint of document input;
- Transformer structure perspective: we explored the effectiveness of different mainstream Transformer structures;
- The significance of encoder and decoder perspective: we designed empirical studies by adding noises on top of the encoder and decoder;
- Training strategy perspective: we reorganized the source documents and include self-supervised learning;
- Summary generation perspective, we explored the uncertainties when repetition problems occur in the summary generation process.

The primary distinction between SDS and MDS lies in the variance of source document numbers. One straightforward way that convert MDS to SDS is simply concatenating text spans and processing them as a flat sequence (Liu et al., 2018a; Chu and Liu, 2019; Brazinskas, Lapata, and Titov, 2020; Mao et al., 2020; Zhao et al., 2022). To aid the models in detecting and modeling document-to-document relationships, a straightforward way is to utilize special tokens as document separators (Fabbri et al., 2019a; Caciularu et al., 2021; Xiao et al., 2022). However, there is no work exploring the impact of document separators qualitatively and quantitatively. This motivated us to analyze whether these special separators help improve models' performance and make the MDS models aware of the document boundaries in the feature space. We conducted the experiments on three Transformer structures and observed that the effect of special tokens is different on models with different hierarchies. Uncertainty analysis is a pivotal approach employed in the examination and assessment of generation systems (Xu, Desai, and Durrett, 2020) which can serve as an important indicator to show how the model performs during the summary generation. We then investigated the variation of summary prediction uncertainty by exploring the relations between separators and the predictive uncertainty of the structures. Certainly, measuring uncertainty in the context of summarization can provide insights into how the presence of document separators affects the behavior of Transformer-based models and their summarization outcomes. By quantifying uncertainty through the entropy calculations, we gained a deeper understanding of the level of confidence or ambiguity the model has in its generated summaries.

Instead of simply concatenating all the input documents into a flat sequence and applying SDS models, the hierarchical Transformer structure (Liu and Lapata, 2019a; Pasunuru et al., 2021b; Li et al., 2020b) has been proposed to specifically solve MDS tasks. This structure has been used for encoding multiple documents in a hierarchical manner, enabling the capture of cross-document relations through the utilization of an attention mechanism. The hierarchical Transformer structure contains a low-level Transformer that encodes tokens and a high-level Transformer that is used to encode coarser-grained textual units. This motivated us to further explore the influence of different hierarchies on MDS performances. We explored the effect of different granularity of high-level Transformer on the performance of MDS models. In this thesis, we considered sentence-level and document-level features as different granularities. Based on the empirical studies, our findings indicate that for MDS tasks involving relatively short documents, flat Transformer models are a suitable choice. Also, the hierarchical structure prefers higher granularity in high-level Transformer structures.

In addition to exploring the hierarchical structure of Transformer-based MDS models, we explored the Transformer's internal structure. Based on the existing

Transformer-based MDS methods, we found that many of the MDS models focus on modifying the components of encoder (Liu and Lapata, 2019a; Pasunuru et al., 2021b; Liu et al., 2021) and fewer works pay attention to ameliorating the decoder (Jin and Wan, 2020; Liu et al., 2022) to cater the requirements for MDS tasks. This motivated us to explore the sensitivity of the components of the encoder-decoder structure. Therefore, we added Gaussian noise at the parameter space of the encoder or decoder to fulfill this purpose. The experiments demonstrate the decoder is more sensitive than the encoder in MDS, which provides a future direction for the research community to pay more attention to the decoder.

Based on the analysis of Transformer-based MDS models, we also payed attention to exploring different training strategies for further enhancing the performance of MDS models. Different training strategies offer unique approaches to utilize available data and optimize model performance. By investigating diverse training strategies, we aimed to identify the most effective methods for training MDS models, leveraging the characteristics of the dataset and the summarization task at hand. These strategies involve using pseudo datasets, fine-tuning on original datasets, or a combining of both. To generate pseudo data, we treated individual documents in a document set as pseudo-summaries and create multiple sets of pseudo-document-summary pairs. We evaluated three training approaches: training exclusively on the pseudo dataset, mixing the pseudo dataset with the original dataset, and a two-step process of training on the pseudo dataset followed by fine-tuning on the original dataset. The experimental results demonstrate that the pretrain-finetune strategy consistently outperformed the other training strategies, leading to improved summarization quality. The analysis of feature distributions further supported this finding, highlighting the alignment between the finetuned model and the baseline model. These results provide valuable insights into the effectiveness of the pretrain-finetune approach in enhancing summarization performance. The findings of this study can guide future research and development in the field of abstractive summarization, emphasizing the importance of training strategies for achieving higher-quality summaries.

Moreover, while the different Transformer structures and training strategies demonstrated variations in performances, an observation is the presence of repetitive patterns in the generated summaries, indicating a potential issue that needs to be addressed in abstractive summarization systems. Salkar et al. (Salkar et al., 2022) noted that the repetition behavior training source has relations with the training source. Liu et al. (Liu et al., 2023) gave two possible reasons behind the repetition problem in abstractive summarization: (1) attending to the same location in the source and

(2) attending to similar but different sentences in the source. In this thesis, we explored the cause of repetitive problems in abstractive summarization by examining predictive uncertainty. We quantified uncertainty scores at each time slot during the summary generation process. The analysis aims to observe how the uncertainty score changes when repetition phenomena occur, allowing us to identify positions where uncertainty is localized in repetitive behavior. The analysis revealed that as the model generates repetitive sentences or words, the uncertainty score rises, pointing out decreased confidence and increased uncertainty regarding the appropriateness and relevance of repeated elements in the summary. Understanding this relationship allowed us to develop strategies to mitigate repetition and improve the quality of generated summaries.

5.2 Methodology

In this section, we introduced how to design the MDS experiments from the following angles: input data, Transformer structures, training strategies and summary generation. Therefore, we designed five experiments to evaluate the behaviors of Transformer-based MDS models: (1) the measurable impact of document separators; (2) the effectiveness of different Transformer structures; (3) the sensitivity of encoder and decoder against noises; (4) different training strategies; (5) repetition in document generation.

5.2.1 The Measurable Impact of Document Separators

We explored if there is a measurable impact in having a document separator between source documents for Transformer-based models. The source documents are separated by special tokens. We modified the source documents to the format of: $D = \{d^1, sep, d^2, sep, ..., sep, d^N\}$, where N is the number of documents in a document set D, the superscript d^n represents the n-th document in the set, and $\mathbf{sep}$ denotes the special tokens. We investigated different Transformer models on two MDS datasets and eleven evaluation metrics to explore the impact of the document separators qualitatively and quantitatively. We also compared and analyze the embedding space of the tokens after they feed into the encoder with and without document separators.

We analyzed and compare the prediction uncertainty from different datasets and different format of source documents by inspecting entropy values during summary generation. We aimed to understand how decisions by adding document separator is reflected in the model's uncertainty. In the generation process, each predictive

position $\mathbf{X}_i$ has an outcome probabilistic distribution $\mathbf{x}_{i1}, ..., \mathbf{x}_{im}$, m is the number of a corpus pool. We used entropy as uncertainty measurement which can be calculate as follow:

$$H(\mathbf{X}_i) = -\sum_{j=1}^{m} P(\mathbf{x}_{ij})\, logP(\mathbf{x}_{ij}) \tag{5.1}$$

Because the size of corpus pool is large and the prediction distribution is usually long-tailed (Xu, Desai, and Durrett, 2020), we sorted the prediction distribution $\mathbf{X}_i$ in descending order and get a minimal set of tokens where the sum prediction values is larger than 0.95, and then normalize the distribution. We calculated the entropy value based on the new distribution $P'(\mathbf{x}_{ij})$. The utilization of entropy as a measure allows us to gauge the distribution of probabilities across different tokens within the predictive positions of the summaries. Higher entropy values indicate a wider spread of probabilities, suggesting that the model is less certain about the most appropriate token to choose. Conversely, lower entropy values suggest that the model is more confident in its token predictions. The quantification of uncertainty through entropy measurements and its qualitative analysis enables us to assess how the introduction of document separators influences the performance of the summaries generated by Transformer-based models. This holistic approach helped us unravel the nuanced impact of document separators on the MDS process and gain valuable insights into the behavior of these models in handling multiple documents inputs.

5.2.2 The Effectiveness of Different Transformer Structures

Transformer structures have become an essential component of many state-of-the-art natural language processing models. However, the design of the Transformer architecture can vary dramatically, and different structures may impact the performance of the model on different tasks. In this study, we aimed to evaluate the effectiveness of different Transformer structures for multi-document summarization tasks. Specifically, we focused on two types of structures: flat Transformer and hierarchical Transformer.

The flat Transformer consists of a single layer of self-attention and feed-forward neural network layers that process the input tokens sequentially. In contrast, the hierarchical Transformer has a more complex structure, where the input tokens are first group into sentences or documents, and then process by local and global Transformer layers. To explore the hierarchical Transformer structure, we investigated two different granularities of high-level Transformer: sentence-level and document-level. Building on the work of Liu (Liu and Lapata, 2019b), we removed the graph structure of the Hierarchical Transformer (HT) model and make modifications to the local

Transformer layers to encode individual sentences or documents. The global Transformer layers are then able to exchange information at the sentence or document level.

Our analysis is motivated by the need to better understand how different Transformer structures can impact the performance of multi-document summarization models. By comparing the performance of the flat Transformer and hierarchical Transformer structures, we aimed to identify which structure is more effective for multiple document summarization data.

5.2.3 The Sensitivity of Encoder and Decoder

In summarization tasks, the encoder plays a crucial role in extracting representations from the input text, while the decoder is responsible for generating the output summary, which requires producing coherent and meaningful language. Given the intricate nature of summary generation, the decoder's role demands fine-grained control and precision, making it potentially more sensitive than the encoder. To explore the sensitivity of encoder-decoder against noises in Transformer-based summarization models, we added Gaussian noise at the parameter space of the encoder or decoder. We devised this experiment based on the intuition that a module (whether it's the encoder or decoder) exhibits varying sensitivity to noise, thereby signifying the differing degrees of importance each module holds for overall performance. Formally, we have:

$$\mathbf{z} = f(\mathbf{x}; \Theta + \alpha\mathbf{n}), \mathbf{n} \sim N(\mu, \delta) \tag{5.2}$$

where $f(\cdot)$ is the component in Transformer; Θ is the parameters in $f(\cdot)$; $\mathbf{n}$ represents Gaussian noise; μ, δ are mean and variance in the Gaussian noise, α is the weighted factor, and $\mathbf{z}$ is the corresponding output.

5.2.4 Different Training Strategies

In this study, we aimed to investigate the impact of different training strategies on Transformer models for abstractive summarization. While we previously examined the components of Transformer models, the specific influence of training strategies remains unexplored. Our objective is to identify the most effective training strategies by leveraging the inherent characteristics of MDS datasets, without the need for external data sources. To create pseudo data utilizing the characteristic multi-document summarization, we adopted a straightforward approach. We treated one document from a given document set as a pseudo-summary, while considering the remaining documents as input documents. This process is iterated, systematically selecting each

document in the set as a pseudo-summary, until all input documents have served as pseudo-summaries. Consequently, we generated multiple sets of pseudo document-summary pairs, which we referred to as pseudo MDS dataset. The original MDS dataset is denoted as the original datasets in the subsequent analysis.

To evaluate the effectiveness of different training strategies, we designed three distinct approaches. Firstly, we trained the MDS model exclusively on the pseudo dataset. Secondly, we mixed the pseudo dataset with the original dataset, creating a comprehensive mega dataset, on which the MDS model is trained. Lastly, we employed a two-step process, initially training the model on the pseudo dataset and subsequently fine-tuning it on the original dataset.

5.2.5 Repetition in Document Generation

For abstractive summarization, a persistent challenge arises from the inclination of models to produce repetitive sentences or words during the summarization process. This tendency creates a loop that is difficult to break, hampering the generation of accurate summaries. To analyse what may cause repetitive problem, we delved into an analysis of prediction uncertainty, examining uncertainty scores throughout the generation process and localizing uncertainty to certain positions in a repetition behavior.

To quantify uncertainty, we employed Equation 5.1, which calculates the uncertainty score for each time slot during the summarization generation. By applying this equation, we obtained a measure of uncertainty that corresponds to the level of doubt or ambiguity associated with the generated output. The analysis focuses on observing how the uncertainty score evolves in response to the occurrence of repetition phenomena.

5.3 Settings for Empirical Studies

In this study, we evaluated the performance of three Transformer models: Vanilla Transformer (VT) (Vaswani et al., 2017), Vanilla Transformer with copy mechanism (VTC), and modified Hierarchical Transformer (HT) (Liu and Lapata, 2019b). These models are assessed on two widely used Multi-Document Summarization (MDS) datasets: Multi-XScience (Lu, Dong, and Charlin, 2020c) and Multi-News (Fabbri et al., 2019a). To comprehensively analyze their performance, we employed eleven evaluation metrics. In the following section, we will introduce these three Transformer models, provide an overview of the datasets, and describe the evaluation metrics utilized in our study.

5.3.1 Summarization Models

Vanilla Transformer (VT) (Vaswani et al., 2017) is a sequence-to-sequence model that is proposed for machine translation task. It is subsequently generalized in various tasks of NLP due to its strong performance (Lin et al., 2021).

Vanilla Transformer with Copy Mechanism (VTC)[1]**.** This variant has a mechanism to copy the attention distribution that one of the randomly chosen attention heads from the encoder side into the decoder, so that the generated text becomes less repetitive and less factually inaccurate.

Hierarchical Transformer (HT) (Liu and Lapata, 2019b) proposed hierarchical attention structure to attend long sequences effectively and capture cross-paragraph contextual relationships. The local Transformer layers encode individual paragraphs and global Transformer layers exchange paragraph-level information from local layers across paragraphs.

5.3.2 Datasets

The empirical studies are based on two widely used MDS datasets: Multi-XScience (Lu, Dong, and Charlin, 2020c) and Multi-News (Fabbri et al., 2019a). Multi-XScience contains data from scientific articles. The task of this dataset is to generate the related work section of a target paper based on its abstract and the abstracts of the articles it refers to. Multi-News collect news articles from the site "newser.com." Each set of source documents has a professionally written summary and the task is to generate that summary based on the sources. Table 5.1 describes the statistics of these two datasets, including the size of the train, test, validation set, the average document length, and the average summary length.

Datasets	Train/ Test/ Validation	Average Document Length	Average Summary Length
Multi-XScience	30,369 / 5,093/ 5,066	778.08	116.44
Multi-News	44,972 / 5,622 / 5,622	2,103.49	263.66

TABLE 5.1. Description of Multi-News and Multi-XScience datasets.

5.3.3 Data Processing

For Multi-XScience and Multi-News datasets, the source documents are separated by a special token named “story_separator_special_tag”. The length of the input

[1]We implemented the VT and VTC based on https://github.com/Alex-Fabbri/Multi-News/tree/master/code/OpenNMT-py-baselines.

documents are restricted to 1024 tokens. In each document set, the number of tokens for one document is $\frac{1024}{N}$, where N is the number of documents in a document set. For some shorter documents, the documents repeat themselves to fill the 1024 token quota. In the Multi-XScience dataset, the citations in the sources and targets are replaced by a common token '@cite'.

5.3.4 Evaluation Metrics

ROUGE[2] Recall-Oriented Understudy for Gisting Evaluation (Lin, 2004b) is a set of evaluation metrics for comparing the overlapping textual units between generated summaries and gold summaries, including ROUGE-1 (R-1), ROUGE-2 (R-2), ROUGE-L (R-L), ROUGE-SU (R-SU). R-1 and R-2 measures the overlapping unigrams and bigrams respectively while R-L identifies the longest co-occurring sequence of n-grams. R-SU is calculated as a statistic to measure the co-occurrence of unigram and skip-bigram.

ROUGE-WE (R-WE) (Ng and Abrecht, 2015) is a variant of the ROUGE metric which replaces the hard lexical matching in ROUGE-N by a soft matching based on the cosine similarity of word embeddings. The soft matching in ROUGE-WE provides a more forgiving evaluation by not strictly requiring exact lexical matches, thus allowing for variations in word order and phrasing.

BLEU BiLingual Evaluation Understudy (Papineni et al., 2002) introduces a brevity penalty term and computes the geometric average of the modified n-gram precision.

S^3 (Peyrard, Botschen, and Gurevych, 2017) is a model-based metric that considers the features from other evaluation metrics, including R-N, R-L, R-WE and JS-divergence, to produce pyramid (pyr) and responsiveness (resp) scores.

BertScore (BS)[3] (Zhang et al., 2020b) measures the soft overlap of the token BERT embeddings from the machine generated summaries and gold summaries.

Relevance (Rel) (Peyrard, 2019) calculates cross-entropy over individually constructed probability distributions for a summary S and a source D using their own semantic units ω: $Relevance(S, D) = \sum_{\omega_i} P_S(\omega_i) \; . \; log(P_D(\omega_i))$, where probability distributions of summary and source document are given by P_S and P_D respectively.

Redundancy(Red) (Peyrard, 2019) evaluates the quality of the accumulation of information in the candidate summaries: $Redundancy(S) = \sum_{\omega_i} P_S(\omega_i) \; . \; log(P_S(\omega_i))$.

[2]The parameters of ROUGE are -c 95 -2 -1 -U -r 1000 -n 4 -w 1.2 -a -m.

[3]The model type of BertScore is bert-base-uncased.

5.4 Empirical Studies and Analyses

This section presented a comprehensive analysis and evaluation of our study's findings in Transformer-based multi-document summarization models. Through rigorous experimentation and quantitative assessments, we explored several crucial aspects related to document separation techniques, Transformer structures, the sensitivity of encoder and decoder components, training strategies, and the relationship between repetition and uncertainty in generated summaries. By examining these aspects, we aimed to provide valuable insights into the effectiveness and performance of different approaches in the field of summarization.

5.4.1 Impact of Document Separators

We investigated the VT, VTC, and HT models on both datasets and eleven evaluation metrics to explore the impact of the document separators. From Table 5.2, interestingly, we found out adding separators reduces models' performance in half of the cases (3 out of 6). For example, model VT with separators performs relatively worse on Multi-News dataset (the results of 8 evaluation metrics are worse among 11 evaluation metrics); model VTC performs relatively worse on both Multi-XScience dataset (the results of 9 evaluation metrics are worse among 11 evaluation metrics) and Multi-News dataset (the results of 8 evaluation metrics are worse among 11 evaluation metrics) when with separators.

These results indicate input document with separators are not very helpful for flat Transformer models. However, we can perceive that the HT model achieves better performance on both datasets with document separators. The discovery is also confirmed by t-SNE visualization (Figure 5.1). After token representations feed into the Transformer encoder, the cluster boundaries of documents with separators are easier to be identified in the embedding space. Potentially, the hierarchical Transformer prefers more structural information of documents to compose the final summaries, while the flat Transformer does not.

Another interesting finding is the most commonly used ROUGE, in a few cases, show the opposite result from other evaluation metrics. For instance, on Multi-XScience dataset, the VT (with document separators) shows better ROUGE results than VT (without document separators) but contradicts the results on "R-WE", "BLEU", "S3", "BertScore", "Redundancy" and "Relevance". It indicates that the ROUGE-centric evaluation system needs to be updated and the measurement of summarization can not rely solely on ROUGE.

We also discovered the relations between document separators and tokens uncertainty scores. Figure 5.2 shows the uncertainty scores of generated tokens of VTC

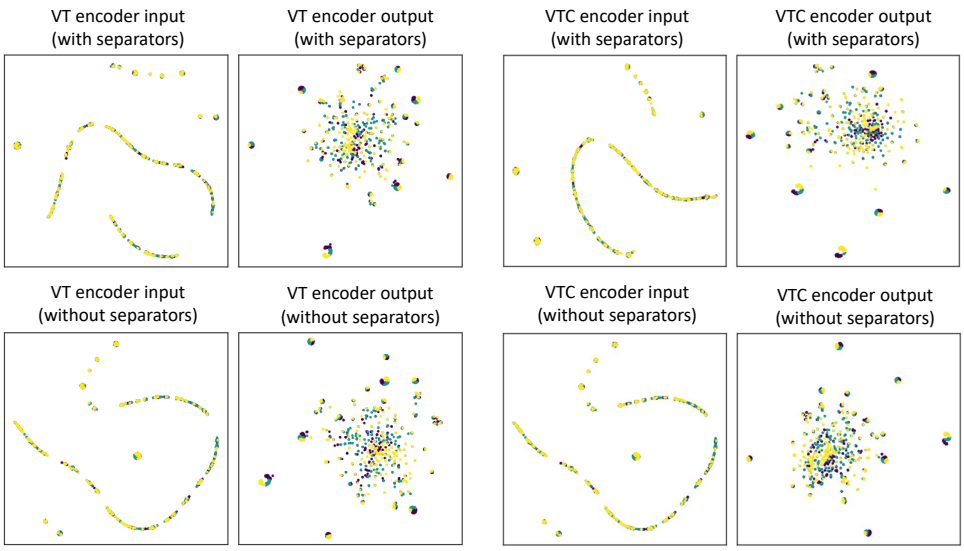


FIGURE 5.1. t-SNE visualization of two embedding space on Multi-News dataset with VT, VTC and HT models: (1) token representations before feeding into the Transformer encoder; (2) token representations after feeding into the Transformer encoder. The figures in the 1st row are the visualization with document separators and in the 2st row are the visualization without document separators.

Datasets	Models	R-1↑	R-2↑	R-L↑	R-SU↑	R-WE↑	BLEU↑	S^3 (pyr/resp)↑	BS↑	Red↓	Rel↑
Multi -XScience	VT	0.2714	0.0490	0.1030	0.0784	0.1523	2.9773	0.2103/0.3609	0.5330	-4.0712	-5.8352
	VT w/o S	0.2670	0.0480	0.1553	0.0767	0.1580	3.3623	0.2202/0.3663	0.5405	-6.1908	-4.8609
	VTC	0.2635	0.0483	0.1499	0.0734	0.1659	4.6037	0.2561/0.3885	0.5590	-7.0585	-4.5802
	VTC w/o S	0.2713	0.0468	0.1502	0.0780	0.1702	4.7615	0.2554/0.3861	0.5621	-7.8402	-4.2908
	HT	0.2571	0.0483	0.1615	0.0692	0.1407	7.1501	0.1769/0.3473	0.5303	-4.6987	-8.0379
	HT w/o S	0.2216	0.0376	0.1446	0.0521	0.1100	5.2862	0.1428/0.3295	0.5108	-4.0142	-11.6068
Multi -News	VT	0.2445	0.0523	0.1301	0.0603	0.1480	2.0054	0.1380/0.3212	0.4622	-5.7674	-7.4220
	VT w/o S	0.2555	0.0550	0.1347	0.0651	0.1491	2.0193	0.1384/0.3214	0.4605	-5.2098	-8.0488
	VTC	0.4233	0.1471	0.2059	0.1625	0.2860	11.3861	0.3778/0.4871	0.5955	-6.0966	3.9027
	VTC w/o S	0.4363	0.1555	0.2053	0.1698	0.2885	13.015	0.3967/0.5017	0.5916	-6.2869	3.8355
	HT	0.2349	0.0371	0.1352	0.0598	0.1154	3.5434	0.1097/0.3074	0.4987	-5.0249	-17.1520
	HT w/o S	0.2304	0.0384	0.1430	0.0580	0.1193	3.0499	0.1023/0.3031	0.4966	-4.9433	-16.8205

TABLE 5.2. Evaluation results on Multi-XScience and Multi-News dataset, both with and without the document separators. "S" indicates document separators. "R-1", "R-2", "R-L", "R-SU","R-WE", "BS", "Red", "Rel" represent ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-SU, ROUGE-WE, BertScore, Redundancy and Relevance."pyr" and "resp" in S^3 are pyramid and responsiveness scores. The upward arrow (↑) signifies that higher values are indicative of better performance, while the downward arrow (↓) implies the opposit.

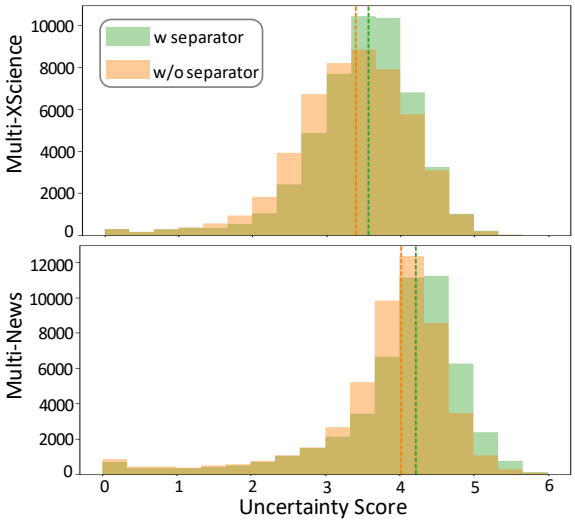


FIGURE 5.2. The uncertainty scores of VTC models on Multi-News and Multi-XScience dataset. The x-axis and y-axis are the value of uncertainty scores and the number of tokens.

models on both datasets. Surprisingly, the figure reflects that separators are associated with high uncertainty score actions which means the separators increase the predictive uncertainty of models. Possible because the separators have no semantic relations with the sources documents and separators may be regarded as noise to increase the predictive uncertainty. The median uncertainty score of Multi-News dataset are larger than the Multi-XScience dataset aligning with the size of datasets.

5.4.2 Quantitative Performance on Different Transformer Structures

We investigated (1) the effectiveness of different Transformer architectures for MDS: flat Transformer and hierarchical Transformer; (2) the influences of different granularities within hierarchical Transformer structure. The results are shown in Table 5.2. In most evaluation metrics, the modified HT model (remove the graph structure) can not achieve as good results as two flat Transformer models on both datasets. The two potential reasons are: (1) the pipeline of the HT model is longer than the flat Transformer models which makes HT model hard to train. (2) the Multi-XScience and Multi-New datasets are not long document summarization dataset. The average document length of Multi-XScience and Multi-New are 778.08 and 2103.49. From the experimental results, we concluded that the HT model is more suitable for lengthy documents. This suggests that for MDS tasks with relatively short documents, flat Transformer models are a good choice to be chosen.

As mentioned in section 5.2.2, to evaluate the influences of different granularities within hierarchical Transformer structure, we removed the graph structure of the

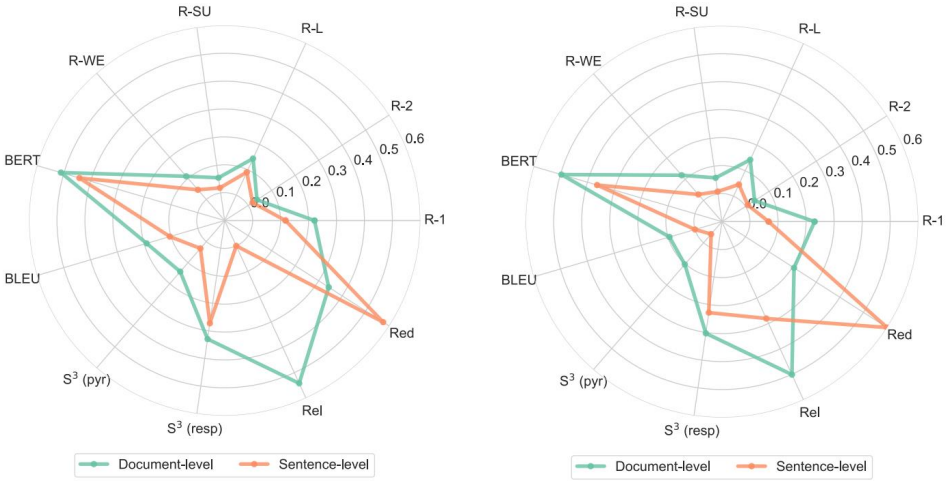


FIGURE 5.3. Performance variation with document-level (green line) and sentence-level (orange line) HT models on Multi-XScience and Multi-News datasets. BLEU, Redundancy and Relevance are scaled (0 to 0.6) to make all point in the plot boundary.

Hierarchical Transformer (HT) model and modified the local Transformer layers to encode individual sentences or documents. Figure 5.3 shows the performances of document-level and sentence-level HT models. All the metrics are showing better performances with the document-level HT compared to the sentence-level HT as the green line exceeds the boundary of the orange line in every dimension (redundancy is the lower the better). The apparent trend implies that a higher level of granularity is more favorable for the hierarchical Transformer structure.

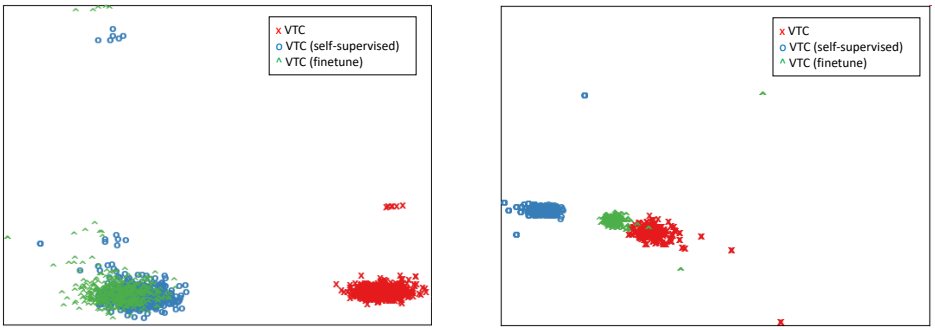


FIGURE 5.4. The feature visualization of VTC, VTC with self-supervised training and VTC with finetuning after self-supervised training with Principal Component Analysis (PCA).

5.4.3 Quantitative Performance on the Sensitivity of Encoder and Decoder

To investigate the hypothesis in section 5.2.3, we selected the VTC model as the foundation for evaluating the effectiveness of the encoder-decoder structure on the Multi-XScience and Multi-News datasets. By examining Table 5.3, we observed large differences in performance when introducing noise to the encoder and decoder in highly noisy scenarios (with α = 1e-1 and α = 1e-2). Specifically, in noisy conditions, we found that adding noise to the decoder has a more substantial impact on performance compared to adding noise to the encoder. However, as the noise levels decreased, the performance gaps between the two approaches narrowed. This observation supports our initial hypothesis that the decoder is more sensitive than the encoder. The potential reasons are: (1) errors or inaccuracies in the decoder can have a cascading effect on subsequent tokens generated during decoding. This error propagation phenomenon can make the decoder more sensitive to small perturbations, as any mistakes or noise introduced during decoding can amplify and affect the overall quality of the generated summary; (2) Transformer-based models often employ an attention mechanism that allows the decoder to focus on different parts of the encoded input during the decoding process. The decoder's sensitivity is crucial in effectively attending to relevant information, and even slight perturbations in the encoded input can impact the attention weights and subsequently influence the decoding process. Consequently, it underscores the crucial role played by the decoder in summarization tasks. These findings shed light on the high importance of the decoder's contribution to the overall summarization process.

5.4.4 Quantitative Performance of Different Training Strategies

The experimental results presented in Table 5.4 provides an overview of the performance of VTC model trained using different pretraining strategies on the Multi-XScience and Multi-News datasets. In the table, the VTC is trained on the original document set and gold summary pairs. The "finetune" strategy refers to the training of the model on the pseudo dataset (introduce in section 5.2.4) first and then fine-tuning on the original dataset. The "self-supervised" strategy denotes training the VTC model exclusively on the pseudo dataset. The "mix" strategy indicates training the model using a combination of the pseudo dataset and the original dataset. By comparing the results obtained from these different training strategies, we aimed to identify the most effective approach for each dataset.

Datasets	Models	R-1↑	R-2↑	R-L↑	R-SU↑	R-WE↑	BLEU↑	S^3 (pyr/resp)↑	BS↑	Red↓	Rel↑
Multi -XScience	En (α=1e-3)	0.2656	0.0477	0.1507	0.0739	0.1660	4.6288	0.2560/0.3881	0.5593	-5.2615	2.5252
	De (α=1e-3)	0.2637	0.0483	0.1499	0.0735	0.1676	4.8116	0.2573/0.3890	0.5608	-5.2806	2.5377
	En (α=1e-2)	0.2433	0.0412	0.1386	0.0650	0.1523	4.0228	0.2276/0.3713	0.5506	-5.2222	2.4878
	De (α=1e-2)	0.2130	0.0362	0.1277	0.0512	0.1333	2.6732	0.1933/0.3535	0.5189	-4.5961	2.4406
	En (α=1e-1)	0.0305	0.0019	0.0232	0.0035	0.0057	0.0979	-0.0786/0.2085	0.3631	-1.8267	0.1420
	De (α=1e-1)	0.0282	0.0039	0.0259	0.0019	0.0115	0.4215	-0.0350/0.2347	0.3533	-0.9935	1.2109
Multi -News	En (α=1e-3)	0.4178	0.1439	0.2063	0.1598	0.2817	10.5326	0.3345/0.4623	0.5943	-5.8867	3.8567
	De (α=1e-3)	0.4172	0.1427	0.2053	0.1589	0.2802	10.6737	0.3348/0.4625	0.5941	-5.8923	3.8533
	En (α=1e-2)	0.2899	0.0689	0.1405	0.0888	0.2095	5.5596	0.2260/0.3778	0.5335	-5.2695	3.7854
	De (α=1e-2)	0.2248	0.0602	0.1134	0.0706	0.1842	4.0850	0.2288/0.3793	0.4972	-4.7247	3.3888
	En (α=1e-1)	0.0938	0.0049	0.0724	0.0101	0.0266	0.0549	-0.0499/ 0.2223	0.3330	-1.2151	1.7586
	De (α=1e-1)	0.0458	0.0018	0.0330	0.0041	0.0186	0.0476	-0.0537/0.2207	0.3410	-2.3011	1.3539

TABLE 5.3. Evaluation results on Multi-XScience and Multi-News dataset about the encoder-decoder structure. "S" indicates document separators. "R-1", "R-2", "R-L", "R-SU","R-WE", "BS", "Red", "Rel" represent ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-SU, ROUGE-WE, BertScore, Redundancy and Relevance."pyr" and "resp" in S^3 are pyramid and responsiveness scores. "En", "De" represent encoder and decoder. The upward arrow (↑) signifies that higher values are indicative of better performance, while the downward arrow (↓) implies the opposit.

For the Multi-XScience dataset, the results show that the VTC (pretrain-finetune) strategy outperforms the VTC model trained on the original dataset across most metrics, indicating the effectiveness of the pretrain-finetune strategy in improving summarization quality. On the other hand, the VTC (self-supervised) exhibits lower performance compared to the VTC (pretrain-finetune), suggesting that just self-supervised training not be as effective for this dataset.

Similarly, for the Multi-News dataset, the results indicate that the VTC model achieves good performance across all metrics, with higher scores on the VTC (pretrain-finetune) strategy, showcasing improved summarization quality. Conversely, the VTC (self-supervised) and VTC (mix) strategy yields lower performance compared to the other strategies.

The comparison of these different training strategies reveals that the pretrain-finetune approach consistently leads to better summarization performance compared to the baseline VTC model and other training strategy, highlighting its effectiveness in improving summarization quality.

To find the potential reason why the finetune strategy works well, we visualized the feature distributions of three training strategy: VTC, VTC (self-supervised) VTC (finetune) using Principal Component Analysis (PCA) as illustrated in Figure 5.4. For the Multi-News dataset, the features comes from encoder of the VTC (self-supervised) model and the VTC (finetuning) model exhibit overlapping, while maintaining distance from the plain VTC model. In contrast, for the Multi-XScience dataset, the VTC (finetune) model is more similar to the plain VTC model, but still noticeably distinct from VTC (self-supervised) model. This observation is consistent with the performance results presented in Table 5.4. In the case of the Multi-XScience dataset, finetuning the model after self-supervised training significantly improves the model's performance compared to the VTC model. However, when the model is only pretrained using self-supervised learning, it performs worse than the VTC model. This discrepancy can be attributed to the fact that the features of the finetuned model closely align with the VTC model's distribution since both models possess better representations for the final prediction. Conversely, for the Multi-News dataset, the finetuned model exhibits only marginal improvements over the VTC model. This observation also explains the overlap between features from the finetuned model and the self-supervised model, as finetuning adjusts the feature distribution towards the 'genuine' distribution, albeit to a limited extent.

5.4.5 The Relation Between Repetition and Uncertainty

The analysis of the relationship between repetition and uncertainty are shown in Figure 5.5. In summary #1, where no repetitions occur, the uncertainties of tokens

Datasets	Models	R-1↑	R-2↑	R-L↑	R-SU↑	R-WE↑	BLEU↑	S^3 (pyr/resp)↑	BS↑	Red↓	Rel↑
Multi-XScience	VTC	0.2635	0.0483	0.1499	0.0734	0.1659	4.6037	0.2561/0.3885	0.5590	-7.0585	-4.5802
	VTC (finetune)	0.2955	0.0558	0.1671	0.0879	0.1770	3.9727	0.2569/0.3886	0.5511	-5.0020	2.5824
	VTC(self-supervised)	0.2585	0.0368	0.1471	0.0678	0.1325	1.2885	0.1694/0.3343	0.5173	-5.3546	2.2064
	VTC(mix)	0.2547	0.0350	0.1468	0.0653	0.1324	1.2922	0.1526/0.3246	0.5176	-5.3285	2.1945
Multi-News	VTC	0.4233	0.1471	0.2059	0.1625	0.2860	11.3861	0.3778/0.4871	0.5955	-6.0966	3.9027
	VTC (finetune)	0.4271	0.1509	0.2084	0.1643	0.2886	11.5514	0.3893/0.4960	0.6004	-6.2075	3.9135
	VTC(self-supervised)	0.2724	0.0484	0.1349	0.0738	0.1399	2.8583	0.1281/0.3159	0.4737	-5.5046	2.4027
	VTC(mix)	0.3046	0.0673	0.1485	0.0938	0.1728	5.2611	0.1909/0.3595	0.4979	-5.8684	2.7281

TABLE 5.4. Different training strategies on Multi-News and Multi-XScience datasets. "S" indicates document separators. "R-1", "R-2", "R-L", "R-SU","R-WE", "BS", "Red", "Rel" represent ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-SU, ROUGE-WE, BertScore, Redundancy and Relevance."pyr" and "resp" in S^3 are pyramid and responsiveness scores. The upward arrow (↑) signifies that higher values are indicative of better performance, while the downward arrow (↓) implies the opposit. "finetune" indicates initially training the model on the pseudo dataset and subsequently fine-tuning it on the original dataset. "self-supervised" indicates training the MDS model exclusively on the pseudo dataset. "mix" indicates training the model on the pseudo dataset with the original dataset.

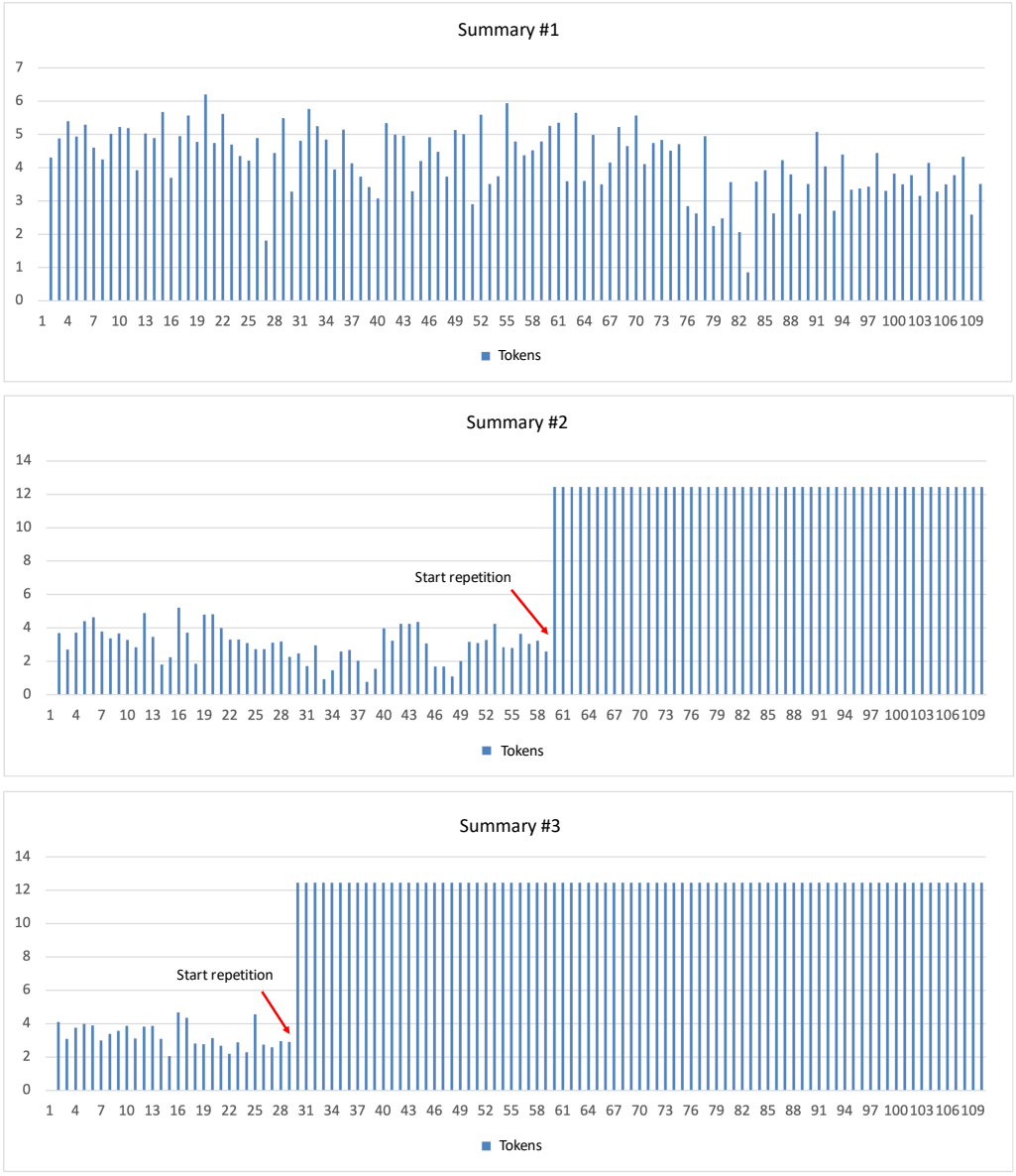


FIGURE 5.5. The relationship between uncertainty scores and token repetitions on different summaries. The X-axis represents the token indexes and Y-axis denotes the uncertainty scores for each token. As shown in the figures, in summary #1, no repetitions happen and the uncertainties of tokens remain in the 'normal' level; however, in summary #2 and #3, the uncertainties climb up very quickly whenever the repetition starts.

remain within a “normal" range. This suggests that the model successfully avoids repetitive patterns, resulting in lower uncertainty scores throughout the summary generation process. Conversely, in summaries #2 and #3, we observed a distinct pattern. As the repetition of tokens or phrases begins, the uncertainty scores escalate rapidly. By comparing uncertainty scores across different time slots, we gained insights into the relationship between repetition and uncertainty in abstractive summarization. When a repetition phenomenon occurs, we observed notable changes in the uncertainty score, indicating a correlation between the two factors. Specifically, as the model generates repetitive sentences or words, the uncertainty score tends to increase. This increase in uncertainty suggests that the model becomes less confident and more uncertain about the appropriateness or relevance of the repeated elements within the summary. By understanding this relationship, we can devise strategies to mitigate repetition and subsequently enhance the quality of generated summaries. By reducing uncertainty through the minimization of repetition, we paved the way for more accurate and reliable abstractive summarization.

5.5 Conclusion and Discussion for the Chapter

This chapter attempts to empirically examine the influences on Transformer behaviors from five important perspectives: document separators, Transformer structures, the sensitivity of encoder-decoder architecture against noises, training strategies, and the relationship between repetition and uncertainty in generated summaries. We first explored the impact of separators on two flat Transformer and one hierarchical Transformer structure. We found that adding separators reduces models' performance for flat Transformer models and increase the predictive uncertainty of models. However, adding separators improve the performance of hierarchical Transformer models. The experiments show that adding separators helps the hierarchical Transformer model aware of the document boundaries while the flat Transformer does not. It indicates that, for models with complex structural information, adding document separators can improve the model performance. The researchers should consider the necessity of applying separators depending on the Transformer structure they use.

The Transformer structure exploring experiments demonstrate that a higher level of granularity is favorable for the hierarchical Transformer structure. The experiments also demonstrate the simple structure, flat Transformer, has been able to show good performance on the Multi-XScience and Multi-News datasets than the complicated hierarchical Transformer structure. The flat Transformer models are good enough for the MDS problems that the length of documents is relatively short.

Furthermore, we had found that adding noise to the decoder has a more pronounced impact on performance compared to adding noise to the encoder. The decoder's sensitivity could be attributed to error propagation during decoding and the attention mechanism's reliance on accurate encoding. These findings highlight the critical role of the decoder in generating high-quality summaries and underscore its significant contribution to the overall summarization process.

The pretrain-finetune strategy that training the model on the pseudo dataset first and then fine-tuning on the original dataset consistently leads to improved summarization performance compared to other training strategies for both the Multi-XScience and Multi-News datasets. This finding highlight the effectiveness of the pretrain-finetune strategy in enhancing the performance of the multi-document summarization models.

Moreover, the analysis of the relationship between repetition and uncertainty provides valuable insights into improving the quality of generated summaries. The findings indicate that as repetition occurs in the summaries, there is a noticeable increase in uncertainty scores. This suggested a correlation between repetition and reduced confidence in the appropriateness and relevance of repeated elements within the summary. By recognizing this relationship, strategies can be developed to mitigate repetition and reduce uncertainty, ultimately enhancing the overall quality of abstractive summaries. These insights contribute to the advancement of abstractive summarization techniques and open avenues for further research in improving the reliability and effectiveness of summary generation.

We also pointed out the possible exploring direction for future MDS work: (1) evaluate the generated summaries from multiple evaluations; (2) add the higher level of granularity information into the models; (3) investigate the MDS method for particularly long input documents; (4) pay more attention to the decoder when design the Transformer-based summarization models; (5) try to reduce the Sudden sharp increase and high uncertainty score during the summary generation process.

Chapter 6

Future Research Directions and Open Issues

Although existing works have established a solid foundation for MDS it is a relatively understudied field compared with SDS and other NLP topics. Summarizing on multi-modal data, medical records, codes, project activities and MDS combining with Internet of Things (Zhang et al., 2020e) have still received less attention. Actually, MDS techniques are beneficial for a variety of practical applications, including generating Wikipedia articles, summarizing news, scientific papers, and product reviews, and individuals, industries have a huge demand for compressing multiple related documents into high-quality summaries. This section outlined several prospective research directions and open issues that we believe are critical to resolving in order to advance the field.

Capturing Cross-document Relations for MDS. Currently, many MDS models still center on a simple concatenation of input documents into a flat sequence, ignoring cross-document relations. Unlike SDS, MDS input documents may contain redundant, complementary, or contradictory information (Radev, 2000). Discovering cross-document relations, which can assist models to extract salient information, improve the coherence and reduce redundancy of summaries(Li et al., 2020b). Research on capturing cross-document relations has begun to gain momentum in the past two years; one of the most widely studied topics is *graphical models*, which can easily be combined with deep learning based models such as graph neural networks and Transformer models. Several existing works indicate the efficacy of graph-based deep learning models in capturing semantic-rich and syntactic-rich representation and generating high-quality summaries (Wang et al., 2020a; Yasunaga et al., 2019; Li et al., 2020b; Yasunaga et al., 2017). To this end, a promising and important direction would be to design a better mechanism to introduce different graph structures (Christensen, Soderland, Etzioni, et al., 2013) or linguistic knowledge (Bing et al., 2015; Ma et al., 2021), possibly into the attention mechanism in deep learning based models, to capture cross-document relations and to facilitate summarization.

Creating More High-quality Datasets for MDS. Benchmark datasets allow researchers to train, evaluate and compare the capabilities of different models at the same stage. High-quality datasets are critical to developing MDS tasks. DUC and TAC, the most common datasets used for MDS tasks, have a relatively small number of samples so are not very suitable for training DNN models. In recent years, some large datasets have been proposed, including WikiSum (Liu et al., 2018a), Multi-News (Fabbri et al., 2019b), and WCEP (Ghalandari et al., 2020a), but more efforts are still needed. Datasets with documents of rich diversity, with minimal positional and extractive biases are desperately required to promote and accelerate MDS research, as are datasets for other applications such as summarization of medical records or dialogue (Molenaar et al., 2020), email (Ulrich, Murray, and Carenini, 2008; Zajic, Dorr, and Lin, 2008), code (Rodeghero et al., 2014; McBurney and McMillan, 2014), software project activities (Alghamdi, Treude, and Wagner, 2020), legal documents (Kanapala, Pal, and Pamula, 2019), and multi-modal data (Li et al., 2020a). The development of large-scale cross-task datasets will facilitate multi-task learning (Xu et al., 2020a). However, the datasets of MDS combining with text classification, question answering, or other language tasks have seldom been proposed in the MDS research community, but these datasets are essential and widely employed in industrial applications.

Improving Evaluation Metrics for MDS. To our best knowledge, there are no evaluation metrics specifically designed for MDS models – SDS and MDS models share the same evaluation metrics. New MDS evaluation metrics should be able to: (1) evaluating the relations between the different input documents in the generated summary; (2) measuring to what extent the redundancy in input documents is reduced; and (3) judging whether the contradictory information across documents is reasonably handled. A good evaluation indicator is able to reflect the true performance of an MDS model and guide design of improved models. However, current evaluation metrics (Fabbri et al., 2021) still have several obvious defects. For example, despite the effectiveness of commonly used ROUGE metrics, they struggle to accurately measure the semantic similarity between a gold and generated summary because ROUGE-based evaluation metrics only consider vocabulary-level distances; as such, even if a ROUGE score improves, it does not necessarily mean that the summary is of a higher quality and so is not ideal for model training. Recently, some works extend ROUGE along with WordNet (ShafieiBavani et al., 2018) or pre-trained LMs (Zhang et al., 2020b) to alleviate these drawbacks. It is challenging to propose evaluation indicators that can reflect the true quality of generated summaries comprehensively and as semantically as human raters. Another frontline challenge for evaluation metrics research is unsupervised evaluation, being explored by a number of recent studies

(Sun and Nenkova, 2019; Gao, Zhao, and Eger, 2020).

Reinforcement Learning for MDS. Reinforcement learning (Mnih et al., 2016) is a cluster of algorithms based on dynamic programming according to the Bellman Equation to deal with sequential decision problems, where state transition dynamics of the environment are provided in advance. Several existing works (Paulus, Xiong, and Socher, 2018; Narayan, Cohen, and Lapata, 2018; Yao et al., 2018) model the document summarization task as a sequential decision problem and adopt reinforcement learning to tackle the task. Although deep reinforcement learning for SDS has made great progress, we still face challenges to adapt existing SDS models to MDS, as the latter suffers from a large state, action space, and problems with high redundancy and contradiction (Mao et al., 2020). Additionally, current summarization methods are based on model-free reinforcement learning algorithms, in which the model is not aware of environment dynamics but continuously explores the environment through simple trial-and-error strategies, so they inevitably suffer from low sampling efficiencies. Nevertheless, the model-based approaches can leverage data more efficiently since they update models upon the prior to the environment. In this case, data-efficient reinforcement learning for MDS could potentially be explored in the future.

Pre-trained Language Models for MDS. In many NLP tasks, the limited labeled corpora are not adequate to train semantic-rich word vectors. Using large-scale, unlabeled, task-agnostic corpora for pre-training can enhance the generalization ability of models and accelerate convergence of networks (Peters et al., 2018; Mikolov et al., 2013). At present, pre-trained LMs have led to successes in many deep learning based NLP tasks. Among the reviewed papers (Zhong et al., 2020; Lebanoff et al., 2019; Li et al., 2020b; Pang et al., 2021; Su et al., 2020; Alambo et al., 2020), multiple works adopt pre-trained LMs for MDS and achieve promising improvements. Applying pre-trained LMs such as BERT (Devlin et al., 2019b), GPT-2 (Radford et al., 2019), GPT-3 (Brown et al., 2020), XLNet (Yang et al., 2019), ALBERT (Lan et al., 2020), or T5 (Raffel et al., 2020), and fine-tuning them on a variety of downstream tasks allows the model to achieve faster convergence speed and can improve model performance. MDS requires the model to have a strong ability to process long sequences. It is promising to explore powerful LMs specifically targeting long sequence input characteristics and avoiding quadratic memory growth for self-attention mechanism, such as Longformer (Beltagy, Peters, and Cohan, 2020), REFORMER (Kitaev, Kaiser, and Levskaya, 2020), or Big Bird (Zaheer et al., 2020) with pre-trained models. Also, tailor-designed pre-trained LMs for summarization have not been well-explored, e.g., using gap sentences generation is more suitable than using masked language model (Zhang et al., 2020a). Most MDS methods focus on

combining pre-trained LMs in encoder and, as for capturing cross-document relations, applying them in decoder is also a worthwhile direction for research (Pasunuru et al., 2021b). Other promising directions in this area involve exploring pre-trained LMs in languages other than English and specialized LMs for dealing with specific summarization tasks, e.g. LMs pre-trained on scientific articles.

Creating Explainable Deep Learning Model for MDS. Researchers are more focused on designing deep architectures towards a certain MDS task by improving the models performance while ignoring their interpretabilities. However, an explainable model can reveal how it generates candidate summaries – to distinguish whether the model has learned the distribution of generating condensed and coherent summaries from multiple documents without bias – and is thus crucial for model building. Recently, a large number of researches into explainable models (Zhang, Nian Wu, and Zhu, 2018; Rudin, 2019) have proposed easing the non-interpretable concern of deep neural networks, within which model attention plays an especially important role in model interpretation (Zhou et al., 2016; Serrano and Smith, 2019). While explainable methods have been intensively researched in NLP (Kumar and Talukdar, 2020; Jain et al., 2020), studies into explainable MDS models are relatively scarce and would benefit from future development.

Adversarial Attack and Defense for MDS. Adversarial examples are strategically modified samples that aim to fool deep neural networks based models. An adversarial example is created via the worst-case perturbation of the input to which a robust DNN model would still assign correct labels, while a vulnerable DNN model would have high confidence in the wrong prediction. The idea of using adversarial examples to examine the robustness of a DNN model originated from research in Computer Vision (Szegedy et al., 2014) and was introduced in NLP by Jia et al. (Jia and Liang, 2017). An essential purpose for generating adversarial examples for neural networks is to utilize these adversarial examples to enhance the model's robustness. Therefore, research on adversarial examples not only helps identify and apply a robust model but also helps to build robust models for different tasks. Following the pioneering work proposed by Jia et al. (Jia and Liang, 2017), many attack methods have been proposed to address this problem in NLP applications (Zhang et al., 2020d) with limited research for MDS (Cheng et al., 2020). It is worth filling this gap by exploring existing and developing new, adversarial attacks on the state-of-the-art DNN-based MDS models.

Multi-modality for MDS. Existing multi-modal summarization is based on non-deep learning techniques (Li et al., 2017a; Jangra et al., 2021; Jangra et al., 2020a; Jangra et al., 2020b), leaving a huge opportunity to exploit deep learning techniques for this task. Multi-modal learning has led to successes in many deep learning tasks,

such as Visual Language Navigation (Wang, Wu, and Shen, 2020) and Visual Question Answering (Antol et al., 2015). Combining MDS with multi-modality has a range of applications:

- text + image: generating summaries with pictures and texts for documents with pictures. This kind of multi-modal summary can improve the satisfaction of users (Zhu et al., 2018);
- text + video: based on the video and its subtitles, generating a concise text summary that describes the main context of video (Palaskar et al., 2019). Movie synopsis is one application;
- text + audio: generating short summaries of audio files that people could quickly preview without actually listening to the entire audio recording (Erol, Lee, and Hull, 2003).

Deep learning is well-suited for multi-modal tasks (Guo, Wang, and Wang, 2019), as it is able to effectively capture highly nonlinear relationships between images, text or video data. Existing MDS models target at dealing with textual data only. Involving richer modalities based on textual data requires models to embrace larger capacity to handle these multi-modal data. The big models such as UNITER (Chen et al., 2020), VisualBERT (Li et al., 2019) deserve more attention in multi-modality MDS tasks. However, at present, there is little multi-modal research work based on MDS; this is a promising, but largely under-explored, area where more studies are expected.

Chapter 7

Conclusion

In this thesis, a collection of innovative techniques for multi-document summarization has been introduced, leveraging deep learning methodologies. These newly devised approaches demonstrate both simplicity and efficacy, as substantiated by their outstanding performance on demanding benchmark datasets.

First, we presented a generic framework to leverage linguistic knowledge to improve the performance of abstractive Transformer-based summarization models. The proposed linguistic guided attention mechanism can be seamlessly incorporated into multiple mainstream Transformer-based summarization models and can outperform existing Transformer-based methods by a large margin. We developed two models based on Flat Transformer (FT) and Hierarchical Transformer (HT). The proposed ParsingSum-HT and ParsingSum-FT incorporate dependency relations with Transformer's multi-head attention for summaries generation. The experiments confirm that utilizing dependency information from the source documents is beneficial to guide the summaries generation process. Based on this work, we presented to encode 45 distinct dependency relations into a dependency relation mask and document positional information for abstractive multi-document summarization. We conducted extensive experiments on two benchmark datasets and the results demonstrate the superior performance of the proposed two encoding methods. The analysis of various settings of the document-aware positional encoding and linguistic-guided encoding can help researchers understand the intuitiveness of the proposed model and could serve as an informative reference to the MDS research community.

Moreover, we also proposed DisentangleSum, a disentangling specificity framework for abstractive multi-document summarization. To optimize the specific feature learning, we applied an orthogonal constraint to encourage the document-specific learner to catch document-specific information. The experiments on two prevalent datasets show the superior performances of the proposed model over other counterparts. Furthermore, we also provided extensive analyses that reveal DisentangleSum exhibits broader coverage of input documents and better preservation of document-related information.

Finally, to examine the behaviors on Transformer based multi-document summarization models, we explored the models from five important perspectives: document separators, Transformer structures, the sensitivity of encoder-decoder architecture against noises, training strategies, and the relationship between repetition and uncertainty in generated summaries. We found that for models with complex structural information, adding document separators can improve the model performance. The researchers should consider the necessity of applying separators depending on the Transformer structure they use. The Transformer structure exploring experiments demonstrate that a higher level of granularity is favorable for the hierarchical Transformer structure. The experiments also demonstrated the flat Transformer models are good enough for the MDS problems that the length of documents is relatively short. Furthermore, we have found that adding noises to the decoder has a more pronounced impact on performance compared to adding noises to the encoder. These findings highlighted the critical role of the decoder in generating high-quality summaries and underscore its significant contribution to the overall summarization process. The pretrain-finetune strategy that trains the model on the pseudo dataset first and then fine-tuning it on the original dataset consistently leads to improved summarization performance when compared to other training strategies. This finding highlighted the effectiveness of the pretrain-finetune strategy in enhancing the performance of the multi-document summarization models. Additionally, the analysis of the relationship between repetition and uncertainty provides valuable insights into improving the quality of generated summaries. By recognizing this relationship, strategies can be developed to mitigate repetition and reduce uncertainty, ultimately enhancing the overall quality of abstractive summaries.

Bibliography

Afantenos, Stergos, Vangelis Karkaletsis, and Panagiotis Stamatopoulos (2005). "Summarization from Medical Documents: A Survey". In: vol. 33. 2, pp. 157–177.

Alambo, Amanuel, Cori Lohstroh, Erik Madaus, Swati Padhee, Brandy Foster, Tanvi Banerjee, Krishnaprasad Thirunarayan, and Michael L. Raymer (2020). "Topic-Centric Unsupervised Multi-Document Summarization of Scientific and News Articles". In: *2020 IEEE International Conference on Big Data (BigData 2020)*. Atlanta, United States, pp. 591–596.

Alghamdi, Mahfouth, 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, pp. 329–342.

Amar, Shmuel, Liat Schiff, Ori Ernst, Asi Shefer, Ori Shapira, and Ido Dagan (2023). "OpenAsp: A Benchmark for Multi-document Open Aspect-based Summarization". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023)*. Singapore, pp. 1967–1991.

Amplayo, Reinald Kim and Mirella Lapata (2021). "Informative and Controllable Opinion Summarization". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2021)*. Online, pp. 2662–2672.

Angelidis, Stefanos 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, pp. 3675–3686.

Antognini, Diego and Boi Faltings (2019). "Learning to Create Sentence Semantic Relation Graphs for Multi-Document Summarization". In: *Proceedings of the 2nd Workshop on New Frontiers in Summarization (NFiS 2019)*. Hongkong, China, pp. 32–41.

Antol, Stanislaw, 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, pp. 2425–2433.

Arora, Rachit 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, pp. 713–718.

Atri, Yash Kumar, Arun Iyer, Tanmoy Chakraborty, and Vikram Goyal (2023). “Promoting Topic Coherence and Inter-Document Consorts in Multi-Document Summarization via Simplicial Complex and Sheaf Graph”. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023)*. Singapore, pp. 2154–2166.

Bahdanau, Dzmitry, 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, United States.

Banerjee, Satanjeev 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*. Ann Arbor, United States, pp. 65–72.

Baralis, Elena, Luca Cagliero, Saima Jabeen, and Alessandro Fiori (2012). “Multi-document Summarization Exploiting Frequent Itemsets”. In: *Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC 2012)*. Riva, Italy, pp. 782–786.

Baxendale, Phyllis B (1958). “Machine-made Index for Technical Literature - An Experiment”. In: vol. 2. 4, pp. 354–361.

Bellot, Patrice, Véronique Moriceau, Josiane Mothe, Eric SanJuan, and Xavier Tannier (2016). “INEX Tweet Contextualization Task: Evaluation, Results and Lesson Learned”. In: vol. 52. 5, pp. 801–819.

Beltagy, Iz, Matthew E Peters, and Arman Cohan (2020). “Longformer: The Long-document Transformer”. In.

Bing, Lidong, Piji Li, Yi Liao, Wai Lam, Weiwei Guo, and Rebecca J. Passonneau (2015). “Abstractive Multi-Document Summarization via Phrase Selection and Merging”. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL 2015)*. Beijing, China, pp. 1587–1597.

Bražinskas, Arthur, Mirella Lapata, and Ivan Titov (2019). "Unsupervised Opinion Summarization as Copycat-Review Generation". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*. Online, pp. 5151–5169.

Brazinskas, Arthur, Mirella Lapata, and Ivan Titov (2020). "Few-Shot Learning for Opinion Summarization". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*. Online, pp. 4119–4135.

Brown, Tom B, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. (2020). "Language Models Are Few-shot Learners". In: *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS 2020)*. Online, pp. 1877–1901.

Caciularu, Avi, Arman Cohan, Iz Beltagy, Matthew E. Peters, Arie Cattan, and Ido Dagan (2021). "CDLM: Cross-Document Language Modeling". In: *Findings of the Association for Computational Linguistics (EMNLP 2021)*. Virtual Event / Punta Cana, Dominican Republic, pp. 2648–2662.

Cao, Qingxing, Xiaodan Liang, Bailin Li, and Liang Lin (2021). "Interpretable Visual Question Answering by Reasoning on Dependency Trees". In: *IEEE transactions on pattern analysis and machine intelligence* 43.3, pp. 887–901.

Cao, Ziqiang, 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, pp. 3053–3059.

Cao, Ziqiang, Furu Wei, Li Dong, Sujian Li, and Ming Zhou (2015a). "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, pp. 2153–2159.

Cao, Ziqiang, Furu Wei, Sujian Li, Wenjie Li, Ming Zhou, and Houfeng Wang (2015b). "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, pp. 829–833.

Carbonell, Jaime G. and Jade Goldstein (1998a). "The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries". In: *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1998)*. Melbourne, Australia, pp. 335–336.

Carbonell, Jaime G. and Jade Goldstein (1998b). "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, pp. 335–336.

Carenini, Giuseppe, 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, pp. 91–100.

Carletta, Jean, Simone Ashby, Sebastien Bourban, Mike Flynn, Maël Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner (2005). "The AMI Meeting Corpus: A Pre-announcement". In: *Machine Learning for Multimodal Interaction, Second International Workshop (MLMI 2005)*. Edinburgh, UK, pp. 28–39.

Chen, Jiaao and Diyi Yang (2020). "Multi-View Sequence-to-Sequence Models with Conversational Structure for Abstractive Dialogue Summarization". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*. Online, pp. 4106–4118.

Chen, Moye, Wei Li, Jiachen Liu, Xinyan Xiao, Hua Wu, and Haifeng Wang (2021a). "SgSum: Transforming Multi-document Summarization into Sub-graph Selection". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*. Online/Punta Cana, Dominican Republic, pp. 4063–4074.

Chen, Yen-Chun, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu (2020). "Uniter: Universal Image-text Representation Learning". In: *Proceedings of 16th European Conference on Computer Vision (ECCV 2020)*. Online, pp. 104–120.

Chen, Yulong, Yang Liu, Liang Chen, and Yue Zhang (2021b). "DialogSumm: A Real-Life Scenario Dialogue Summarization Dataset". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL 2021)*. Online, pp. 5062–5074.

Cheng, Minhao, 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, United States, pp. 3601–3608.

Cho, Sangwoo, 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, pp. 1027–1038.

Christensen, Janara, 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, United States, pp. 1163–1173.

Chu, Eric 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, pp. 1223–1232.

Chung, Junyoung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio (2014). "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling". In: *Proceedings of the 28th Annual Conference on Neural Information Processing Systems Workshop on Deep Learning (NIPS 2014)*. Montreal, Canada.

Coavoux, Maximin, Hady Elsahar, and Matthias Gallé (2019). "Unsupervised Aspect-Based Multi-Document Abstractive Summarization". In: *Proceedings of the 2nd Workshop on New Frontiers in Summarization (NFiS 2019)*. Hong Kong, China, pp. 42–47.

Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa (2011). "Natural Language Processing (Almost) from Scratch". In: vol. 12, pp. 2493–2537.

Deguchi, Hiroyuki, Akihiro Tamura, and Takashi Ninomiya (2019). "Dependency-Based Self-Attention for Transformer NMT". In: *Proceedings of the International Conference on Recent Advances in Natural Language Processing, (RANLP 2019)*. Ed. by Ruslan Mitkov and Galia Angelova. Varna, Bulgaria, pp. 239–246.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019a). "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, Minnesota, pp. 4171–4186.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019b). "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, pp. 4171–4186.

Devlin, Jacob, 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, pp. 1370–1380.

Dong, Li, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon (2019). "Unified Language Model Pre-training for Natural Language Understanding and Generation". In: *Proceedings of the 33th Annual Conference on Neural Information Processing Systems (NeurIPS 2019)*. Vancouver, Canada, pp. 13042–13054.

Dos Santos, Cicero 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, pp. 69–78.

Dozat, Timothy and Christopher D. Manning (2017a). "Deep Biaffine Attention for Neural Dependency Parsing". In: *Proceedings of the 5th International Conference on Learning Representations, (ICLR 2017)*. Toulon, France.

Dozat, Timothy and Christopher D. Manning (2017b). "Deep Biaffine Attention for Neural Dependency Parsing". In: *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*. Toulon, France.

El-Kassas, Wafaa S., Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed (2021). "Automatic Text Summarization: A Comprehensive Survey". In: vol. 165, p. 113679.

Enarvi, Seppo, Marilisa Amoia, Miguel Del-Agua Teba, Brian Delaney, Frank Diehl, Stefan Hahn, Kristina Harris, Liam McGrath, Yue Pan, Joel Pinto, et al. (2020). "Generating Medical Reports from Patient-doctor Conversations Using Sequence-to-sequence Models". In: *Proceedings of the First Workshop on Natural Language Processing for Medical Conversations*. Online, pp. 22–30.

Erkan, Günes and Dragomir R Radev (2004a). "Lexrank: Graph-based Lexical Centrality as Salience in Text Summarization". In: vol. 22, pp. 457–479.

Erkan, Günes and Dragomir R. Radev (2004b). "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization". In: vol. 22, pp. 457–479.

Ernst, Ori, Avi Caciularu, Ori Shapira, Ramakanth Pasunuru, Mohit Bansal, Jacob Goldberger, and Ido Dagan (2022). "Proposition-Level Clustering for Multi-Document

Summarization". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2022)*. Seattle, United States, pp. 1765–1779.

Erol, Berna, 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, pp. 25–28.

Fabbri, Alexander R, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev (2021). "Summeval: Re-evaluating Summarization Evaluation". In: vol. 9, pp. 391–409.

Fabbri, Alexander R., Irene Li, Tianwei She, Suyi Li, and Dragomir R. Radev (2019a). "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, pp. 1074–1084.

Fabbri, Alexander R., Irene Li, Tianwei She, Suyi Li, and Dragomir R. Radev (2019b). "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, pp. 1074–1084.

Fan, Angela, Claire Gardent, Chloé Braud, and Antoine Bordes (2019). "Using Local Knowledge Graph Construction to Scale Seq2Seq Models to Multi-Document Inputs". 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, pp. 4184–4194.

Feng, Xiachong, Xiaocheng Feng, Bing Qin, Xinwei Geng, and Ting Liu (2021). "Dialogue Discourse-Aware Graph Convolutional Networks for Abstractive Meeting Summarization". In: *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI 2021)*. Online, pp. 3808–3814.

Fernandes, Patrick, Miltiadis Allamanis, and Marc Brockschmidt (2019). "Structured Neural Summarization". In: *Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)*. New Orleans, United States.

Ferreira, Rafael, Luciano de Souza Cabral, Frederico Freitas, Rafael Dueire Lins, Gabriel de França Silva, Steven J Simske, and Luciano Favaro (2014). "A Multidocument Summarization System based on Statistics and Linguistic Treatment". In: vol. 41. 13, pp. 5780–5787.

Ganesan, Kavita, 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, pp. 340–348.

Gao, Yang, 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, pp. 1347–1354.

Gao, Yanjun, 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, pp. 404–418.

Gehrmann, Sebastian, Yuntian Deng, and Alexander M. Rush (2018). "Bottom-Up Abstractive Summarization". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*. Brussels, Belgium, pp. 4098–4109.

Gerani, Shima, 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, pp. 1602–1613.

Ghalandari, Demian Gholipour, Chris Hokamp, Nghia The Pham, John Glover, and Georgiana Ifrim (2020a). "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, pp. 1302–1308.

Ghalandari, Demian Gholipour, Chris Hokamp, Nghia The Pham, John Glover, and Georgiana Ifrim (2020b). "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, pp. 1302–1308.

Goldstein, Jade, Vibhu O Mittal, Jaime G Carbonell, and Mark Kantrowitz (2000). "Multi-document Summarization by Sentence Extraction". In: *Proceedings of the 2000 Conference of the North American Chapter of the Association for Computational Linguistics: Applied Natural Language Processing Conference (NAACL-ANLP 2000)*. Seattle, United States, pp. 91–98.

Goodwin, Travis R., Max E. Savery, and Dina Demner-Fushman (2020). "Flight of the PEGASUS? Comparing Transformers on Few-shot and Zero-shot Multi-document Abstractive Summarization". In: *Proceedings of the 28th International*

Conference on Computational Linguistics (COLING 2020). Online, pp. 5640–5646.

Grail, Quentin, Julien Perez, and Eric Gaussier (2021). "Globalizing BERT-based Transformer Architectures for Long Document Summarization". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2021)*. Online, pp. 1792–1810.

Grusky, Max, Mor Naaman, and Yoav Artzi (2018). "Newsroom: A Dataset of 1.3 Million Summaries with Diverse Extractive Strategies". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018)*. New Orleans, United States, pp. 708–719.

Gu, Jiatao, 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, pp. 1631–1640.

Guo, Wenzhong, Jianwen Wang, and Shiping Wang (2019). "Deep Multimodal Representation Learning: A Survey". In: vol. 7, pp. 63373–63394.

Gupta, Vishal and Gurpreet Singh Lehal (2010). "A Survey of Text Summarization Extractive Techniques". In: vol. 2. 3, pp. 258–268.

Haghighi, Aria 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, pp. 362–370.

Halliday, Michael Alexander Kirkwood, Christian MIM Matthiessen, Michael Halliday, and Christian Matthiessen (2014). *An Introduction to Functional Grammar*. Routledge.

Haque, Majharul, Suraiya Pervin, Zerina Begum, et al. (2013). "Literature Review of Automatic Multiple Documents Text Summarization". In: vol. 3. 1, pp. 121–129.

Hirao, Tsutomu, 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, pp. 4177–4186.

Hirao, Tsutomu, Jun Suzuki, Hideki Isozaki, and Eisaku Maeda (2004). "Dependency-based Sentence Alignment for Multiple Document Summarization". In: *Proceedings of 20th International Conference on Computational Linguistics (COLING 2004)*. Geneva, Switzerland, pp. 446–452.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long Short-term Memory". In: vol. 9. 8, pp. 1735–1780.

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White (1989). "Multilayer Feedforward Networks are Universal Approximators". In: vol. 2. 5, pp. 359–366.

Huang, Zhiheng, Wei Xu, and Kai Yu (2015). "Bidirectional LSTM-CRF Models for Sequence Tagging". In.

Im, Jinbae, Moonki Kim, Hoyeop Lee, Hyunsouk Cho, and Sehee Chung (2021). "Self-Supervised Multimodal Opinion Summarization". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*. Online, pp. 388–403.

Jain, Sarthak, 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, pp. 4459–4473.

Jangra, Anubhav, Adam Jatowt, Mohammad Hasanuzzaman, and Sriparna Saha (2020a). "Text-image-video Summary Generation Using Joint Integer Linear Programming". In: vol. 12036, p. 190.

Jangra, Anubhav, Sriparna Saha, Adam Jatowt, and Mohammad Hasanuzzaman (2020b). "Multi-modal Summary Generation Using Multi-objective Optimization". In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020)*. Online, pp. 1745–1748.

Jangra, Anubhav, Sriparna Saha, Adam Jatowt, and Mohammed Hasanuzzaman (2021). "Multi-Modal Supplementary-Complementary Summarization using Multi-Objective Optimization". In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*. Online, pp. 818–828.

Jelinek, Fred, Robert L Mercer, Lalit R Bahl, and James K Baker (1977). "Perplexity - A Measure of the Difficulty of Speech Recognition Tasks". In: vol. 62. S1, S63–S63.

Jia, Robin 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, pp. 2021–2031.

Jin, Hanqi and Xiaojun Wan (2020). "Abstractive Multi-Document Summarization via Joint Learning with Single-Document Summarization". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Findings (ACL 2020)*. Online, pp. 2545–2554.

Jin, Hanqi, Tianming Wang, and Xiaojun Wan (2020a). "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, pp. 6244–6254.

Jin, Hanqi, Tianming Wang, and Xiaojun Wan (2020b). "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, pp. 6244–6254.

Jin, Hanqi, Tianming Wang, and Xiaojun Wan (2020c). "SemSUM: Semantic Dependency Guided Neural Abstractive Summarization". In: *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2020)*. New York, United States, pp. 8026–8033.

Joshi, Anirudh, Namit Katariya, Xavier Amatriain, and Anitha Kannan (2020). "Dr. Summarize: Global Summarization of Medical Dialogue by Exploiting Local Structures". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, (EMNLP 2020)*. Online, pp. 3755–3763.

Kanapala, Ambedkar, Sukomal Pal, and Rajendra Pamula (2019). "Text Summarization from Legal Documents: A Survey". In: vol. 51. 3, pp. 371–402.

Kim, Yoon (2014). "Convolutional Neural Networks for Sentence Classification". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*. Doha, Qatar, pp. 1746–1751.

Kipf, Thomas N. 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.

Kitaev, Nikita, 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.

Koay, Jia Jin, Alexander Roustai, Xiaojin Dai, Dillon Burns, Alec Kerrigan, and Fei Liu (2020). "How Domain Terminology Affects Meeting Summarization Performance". In: *Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020)*. Online, pp. 5689–5695.

Krizhevsky, Alex, 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, pp. 1106–1114.

Kulkarni, Sayali, Sheide Chammas, Wan Zhu, Fei Sha, and Eugene Ie (2020). "AQuaMuSe: Automatically Generating Datasets for Query-Based Multi-Document Summarization". In.

Kumar, Sawan 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, pp. 8730–8742.

Lan, Zhenzhong, 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.

Lebanoff, Logan, 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, pp. 2175–2189.

Lebanoff, Logan, 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, pp. 4131–4141.

LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner (1998). "Gradient-based Learning Applied to Document Recognition". In: vol. 86. 11, pp. 2278–2324.

Leite, Daniel S, Lucia HM Rino, Thiago AS Pardo, and Maria das Graças Volpe Nunes (2007). "Extractive Automatic Summarization: Does more Linguistic Knowledge Make a Difference?" In: *Proceedings of the 2rd Workshop on TextGraphs: Graph-based Algorithms for Natural Language Processing*. Rochester, United States, pp. 17–24.

Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer (2020). "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*. Online, pp. 7871–7880.

Li, Haoran, Peng Yuan, Song Xu, Youzheng Wu, Xiaodong He, and Bowen Zhou (2020a). "Aspect-Aware Multimodal Summarization for Chinese E-Commerce Products". In: *Proceedings of The 34th AAAI Conference on Artificial Intelligence (AAAI 2020)*. New York, United States, pp. 8188–8195.

Li, Haoran, Junnan Zhu, Cong Ma, Jiajun Zhang, and Chengqing Zong (2017a). "Multi-modal Summarization for Asynchronous Collection of Text, Image, Audio and Video". In: *Proceedings of the 2017 Conference on Empirical Methods in*

Natural Language Processing (EMNLP 2017). Copenhagen, Denmark, pp. 1092–1102.

Li, Liunian Harold, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang (2019). "Visualbert: A Simple and Performant Baseline for Vision and Language". In.

Li, Piji, 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 2017)*. Copenhagen, Denmark, pp. 91–99.

Li, Piji, Wai Lam, Lidong Bing, Weiwei Guo, and Hang Li (2017b). "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, pp. 2081–2090.

Li, Wei, Xinyan Xiao, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du (2020b). "Leveraging Graph to Improve Abstractive Multi-Document Summarization". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*. Online, pp. 6232–6243.

Li, Wei, Xinyan Xiao, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du (2020c). "Leveraging Graph to Improve Abstractive Multi-Document Summarization". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*. Online, pp. 6232–6243.

Li, Wei, Xinyan Xiao, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du (2020d). "Leveraging Graph to Improve Abstractive Multi-Document Summarization". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*. Online, pp. 6232–6243.

Lin, Chin-Yew (2004a). "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Proceedings of the Workshop of Text Summarization Branches Out*. Barcelona, Spain, pp. 74–81.

Lin, Chin-Yew (2004b). "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Proceedings of the Workshop of Text Summarization Branches Out*. Barcelona, Spain, pp. 74–81.

Lin, Tianyang, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu (2021). "A Survey of Transformers". In: *CoRR* abs/2106.04554.

Liu, Chunyi, Peng Wang, Jiang Xu, Zang Li, and Jieping Ye (2019). "Automatic Dialogue Summary Generation for Customer Service". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD 2019)*. Anchorage, United States, pp. 1957–1965.

Liu, Peter J., Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer (2018a). "Generating Wikipedia by Summarizing Long Sequences". In: *Proceedings of the 6th International Conference on Learning Representations (ICLR 2018)*. Vancouver, Canada.

Liu, Peter J., Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer (2018b). "Generating Wikipedia by Summarizing Long Sequences". In: *Proceedings of 6th International Conference on Learning Representations (ICLR2018)*. Vancouver, Canada.

Liu, Shuaiqi, Jiannong Cao, Ruosong Yang, and Zhiyuan Wen (2021). "Highlight-Transformer: Leveraging Key Phrase Aware Attention to Improve Abstractive Multi-Document Summarization". In: *Findings of the Association for Computational Linguistics (ACL/IJCNLP 2021)*. Online, pp. 5021–5027.

Liu, Yang and Mirella Lapata (2019a). "Hierarchical Transformers for Multi-Document Summarization". In: *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL 2019)*. Florence, Italy, pp. 5070–5081.

Liu, Yang and Mirella Lapata (2019b). "Hierarchical Transformers for Multi-Document Summarization". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*. Florence, Italy, pp. 5070–5081.

Liu, Yixin, Pengfei Liu, Dragomir R. Radev, and Graham Neubig (2022). "BRIO: Bringing Order to Abstractive Summarization". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)*. Dublin, Ireland, pp. 2890–2903.

Liu, Yizhu, Xinyue Chen, Xusheng Luo, and Kenny Q. Zhu (2023). "Reducing repetition in convolutional abstractive summarization". In: *Nature Language Engineering* 29.1, pp. 81–109.

Louis, Annie and Ani Nenkova (2013). "Automatically Assessing Machine Summary Content Without A Gold Standard". In: vol. 39. 2, pp. 267–300.

Lu, Yao, Yue Dong, and Laurent Charlin (2020a). "Multi-XScience: A Large-scale Dataset for Extreme Multi-document Summarization of Scientific Articles". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*. Online, pp. 8068–8074.

Lu, Yao, Yue Dong, and Laurent Charlin (2020b). "Multi-XScience: A Large-scale Dataset for Extreme Multi-document Summarization of Scientific Articles". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*. Online, pp. 8068–8074.

Lu, Yao, Yue Dong, and Laurent Charlin (2020c). "Multi-XScience: A Large-scale Dataset for Extreme Multi-document Summarization of Scientific Articles". In:

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020). Online, pp. 8068–8074.

Luo, Wencan, 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, pp. 80–85.

Ma, Congbo, Wei Emma Zhang, Mingyu Guo, Hu Wang, and Quan Z Sheng (2020). "Multi-document Summarization via Deep Learning Techniques: A Survey". In: *arXiv preprint arXiv:2011.04843*.

Ma, Congbo, Wei Emma Zhang, Pitawelayalage Dasun Dileepa Pitawela, Yutong Qu, Haojie Zhuang, and Hu Wang (2022). "Document-aware Positional Encoding and Linguistic-guided Encoding for Abstractive Multi-document Summarization". In: *Proceedings of the IEEE Word Congress on Computational Intelligence (WCCI 2022)*. Padua, Italy.

Ma, Congbo, Wei Emma Zhang, Hu Wang, Shubham Gupta, and Mingyu Guo (2021). "Incorporating Linguistic Knowledge for Abstractive Multi-document Summarization". In.

Mani, Inderjeet and Eric Bloedorn (1997). "Multi-Document Summarization by Graph Search and Matching". In: *Proceedings of the 14th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conference (AAAI 1997)*. Providence, United States, pp. 622–628.

Mao, Yuning, Yanru Qu, Yiqing Xie, Xiang Ren, and Jiawei Han (2020). "Multi-document Summarization with Maximal Marginal Relevance-guided Reinforcement Learning". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*. Online, pp. 1737–1751.

McBurney, Paul W 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, pp. 279–290.

Miao, Yishu 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, pp. 319–328.

Mihalcea, Rada and Paul Tarau (2004). "TextRank: Bringing Order into Text". In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*. Barcelona, Spain, pp. 404–411.

Mihalcea, Rada 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, pp. 19–24.

Mikolov, Tomas, 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, pp. 3111–3119.

Mnih, Volodymyr, 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, pp. 1928–1937.

Molenaar, Sabine, 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, pp. 76–88.

Moro, Gianluca, Luca Ragazzi, Lorenzo Valgimigli, and Davide Freddi (2022). "Discriminative Marginalized Probabilistic Neural Method for Multi-Document Summarization of Medical Literature". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL 2022)*. Dublin, Ireland, pp. 180–189.

Nallapati, Ramesh, Feifei Zhai, and Bowen Zhou (2017). "SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents". In: *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI 2017)*. San Francisco, United States, pp. 3075–3081.

Narayan, Shashi, 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, pp. 1747–1759.

Nayeem, Mir Tafseer, 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, pp. 1191–1204.

Nema, Preksha, Mitesh M. Khapra, Anirban Laha, and Balaraman Ravindran (2017). "Diversity driven attention model for query-based abstractive summarization".

In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*. Vancouver, Canada, pp. 1063–1072.

Nenkova, Ani and Kathleen R. McKeown (2012). "A Survey of Text Summarization Techniques". In: *Mining Text Data*, pp. 43–76.

Nenkova, Ani, Rebecca Passonneau, and Kathleen McKeown (2007). "The Pyramid Method: Incorporating Human Content Selection Variation in Summarization Evaluation". In: vol. 4. 2, p. 4.

Nenkova, Ani and Rebecca J Passonneau (2004). "Evaluating Content Selection in Summarization: The Pyramid Method". In: *Proceedings of the 2004 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2004)*. Boston, United States, pp. 145–152.

Ng, Jun-Ping and Viktoria Abrecht (2015). "Better Summarization Evaluation with Word Embeddings for ROUGE". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*. Lisbon, Portugal, pp. 1925–1930.

Palaskar, Shruti, 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, pp. 6587–6596.

Pang, Richard Yuanzhe, Ádám Dániel Lelkes, Vinh Q. Tran, and Cong Yu (2021). "AgreeSum: Agreement-Oriented Multi-Document Summarization". In: *Findings of the Association for Computational Linguistics (ACL-IJCNLP 2021)*. Online, pp. 3377–3391.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2002). "BLEU: A Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*. Philadelphia, United States, pp. 311–318.

Passonneau, Rebecca J, 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, pp. 143–147.

Pasunuru, Ramakanth, Asli Celikyilmaz, Michel Galley, Chenyan Xiong, Yizhe Zhang, Mohit Bansal, and Jianfeng Gao (2021a). "Data Augmentation for Abstractive Query-Focused Multi-Document Summarization". In: *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI 2021)*. Online, pp. 13666–13674.

Pasunuru, Ramakanth, Mengwen Liu, Mohit Bansal, Sujith Ravi, and Markus Dreyer (2021b). "Efficiently Summarizing Text and Graph Encodings of Multi-Document Clusters". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2021)*. Online, pp. 4768–4779.

Paulus, Romain, 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.

Peters, Matthew E., 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, pp. 2227–2237.

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

Peyrard, Maxime, Teresa Botschen, and Iryna Gurevych (2017). "Learning to Score System Summaries for Better Content Selection Evaluation". In: *Proceedings of the Workshop on New Frontiers in Summarization (NFiS@EMNLP 2017)*. Copenhagen, Denmark, pp. 74–84.

Puduppully, Ratish Surendran, Parag Jain, Nancy Chen, and Mark Steedman (2023). "Multi-Document Summarization with Centroid-Based Pretraining". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (ACL 2023)*. Toronto, Canada, pp. 128–138.

Radev, Dragomir R. (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, pp. 74–83.

Radev, Dragomir R, Hongyan Jing, Małgorzata Styś, and Daniel Tam (2004). "Centroid-based Summarization of Multiple Documents". In: vol. 40. 6, pp. 919–938.

Radev, Dragomir R., Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara (2013). "The ACL Anthology Network Corpus". In: vol. 47. 4, pp. 919–944.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever (2019). "Language Models are Unsupervised Multitask Learners". In: vol. 1. 8, p. 9.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2020). "Exploring the Limits of

Transfer Learning with a Unified Text-to-Text Transformer". In: vol. 21, 140:1–140:67.

Rodeghero, Paige, 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, pp. 390–401.

Rudin, Cynthia (2019). "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead". In: vol. 1. 5, pp. 206–215.

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (1986). "Learning Representations by Back-propagating Errors". In: vol. 323. 6088, pp. 533–536.

Sabour, Sara, Nicholas Frosst, and Geoffrey E Hinton (2017). "Dynamic Routing Between Capsules". In: *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS 2017)*. Long Beach, United States, pp. 3856–3866.

Salkar, Nikita, Thomas A. Trikalinos, Byron C. Wallace, and Ani Nenkova (2022). "Self-Repetition in Abstractive Neural Summarizers". In: *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (AACL/IJCNLP 2022)*. Online, pp. 341–350.

Schroff, Florian, Dmitry Kalenichenko, and James Philbin (2015). "FaceNet: A unified embedding for face recognition and clustering". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015)*. Boston, USA, pp. 815–823.

See, Abigail, Peter J. Liu, and Christopher D. Manning (2017a). "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, pp. 1073–1083.

See, Abigail, Peter J. Liu, and Christopher D. Manning (2017b). "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, pp. 1073–1083.

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

ShafieiBavani, Elaheh, Mohammad Ebrahimi, Raymond K. Wong, and Fang Chen (2018). "A Graph-Theoretic Summary Evaluation for Rouge". In: *Proceedings*

of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018). Brussels, Belgium, pp. 762–767.

Shah, Chintan and Anjali Jivani (2016). "Literature Study on Multi-document Text Summarization Techniques". In: *Proceedings of the International Conference on Smart Trends for Information Technology and Computer Communications (SmartCom 2016)*. Jaipur, India, pp. 442–451.

Shapira, Ori, David Gabay, Yang Gao, Hadar Ronen, Ramakanth Pasunuru, Mohit Bansal, Yael Amsterdamer, and Ido Dagan (2019). "Crowdsourcing Lightweight Pyramids for Manual Summary Evaluation". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*. Minneapolis, United States, pp. 682–687.

Shirwandkar, Nikhil S 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, pp. 1–5.

Singh, Abhishek Kumar, 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, pp. 5473–5480.

Smith, Samuel L., Pieter-Jan Kindermans, Chris Ying, and Quoc V. Le (2018). "Don't Decay the Learning Rate, Increase the Batch Size". In: *Proceedings of the 6th International Conference on Learning Representations (ICLR 2018)*. Vancouver, Canada.

Song, Kaiqiang, Logan Lebanoff, Qipeng Guo, Xipeng Qiu, Xiangyang Xue, Chen Li, Dong Yu, and Fei Liu (2020a). "Joint Parsing and Generation for Abstractive Summarization". In: *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2020)*. New York, United States, pp. 8894–8901.

Song, Yan, Yuanhe Tian, Nan Wang, and Fei Xia (2020b). "Summarizing Medical Conversations via Identifying Important Utterances". In: *Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020)*. Online, pp. 717–729.

Song, Yun-Zhu, Yi-Syuan Chen, and Hong-Han Shuai (2022). "Improving Multi-Document Summarization through Referenced Flexible Extraction with Credit-Awareness". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2022)*. Seattle, United States, pp. 1667–1681.

Su, Dan, Yan Xu, Tiezheng Yu, Farhad Bin Siddique, Elham J. Barezi, and Pascale Fung (2020). "CAiRE-COVID: A Question Answering and Query-focused Multi-Document Summarization System for COVID-19 Scholarly Information Management". In: *Proceedings of the 1st Workshop on NLP for COVID-19*. Online.

Sun, Kai, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu (2019). "Aspect-Level Sentiment Analysis Via Convolution over Dependency Tree". 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, pp. 5678–5687.

Sun, Simeng 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, pp. 1216–1221.

Szegedy, Christian, 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.

Takase, Sho, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, and Masaaki Nagata (2016). "Neural Headline Generation on Abstract Meaning Representation". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*. Austin, United States, pp. 1054–1059.

Tan, Haihui, Ziyu Lu, and Wenjie Li (2017). "Neural Network based Reinforcement Learning for Real-time Pushing on Text Stream". In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017)*. Tokyo, Japan, pp. 913–916.

Tas, Oguzhan and Farzad Kiyani (2007). "A Survey Automatic Text Summarization". In: vol. 5. 1, pp. 205–213.

Tu, Zhaopeng, 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, pp. 76–85.

Ulrich, Jan, Gabriel Murray, and Giuseppe Carenini (2008). "A Publicly Available Annotated Corpus for Supervised Email Summarization". In: *Proceedings of the 23th AAAI Conference on Artificial Intelligence in Enhanced Messaging Workshop (AAAI 2008)*. Chicago, United States, pp. 77–82.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). "Attention is All you Need". In: *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NeurIPS 2017)*. Long Beach, USA, pp. 5998–6008.

Vinyals, Oriol, Meire Fortunato, and Navdeep Jaitly (2015). "Pointer Networks". In: *Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS 2015)*. Montreal, Canada, pp. 2692–2700.

Vodolazova, Tatiana, 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, pp. 164–175.

Wan, Xiaojun and Jianwu Yang (2006). "Improved Affinity Graph based Multi-document Summarization". In: *Proceedings of the 2006 Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (NAACL 2006)*. New York,United States, pp. 336–347.

Wan, Xiaojun 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, pp. 299–306.

Wang, Danqing, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang (2020a). "Heterogeneous Graph Neural Networks for Extractive Document Summarization". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*. Online, pp. 6209–6219.

Wang, Hu, 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, pp. 126–141.

Wang, Kai, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang (2020b). "Relational Graph Attention Network for Aspect-based Sentiment Analysis". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*. Online, pp. 3229–3238.

Wang, Lu 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, pp. 47–57.

Wang, Lucy Lu, Yulia Otmakhova, Jay DeYoung, Thinh Hung Truong, Bailey Kuehl, Erin Bransom, and Byron C. Wallace (2023). "Automated Metrics for Medical

Multi-Document Summarization Disagree with Human Evaluations". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023)*. Toronto, Canada, pp. 9871–9889.

Wen, Liang, Houfeng Wang, Yingwei Luo, and Xiaolin Wang (2022). "M3: A Multi-View Fusion and Multi-Decoding Network for Multi-Document Reading Comprehension". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP 2022)*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 1450–1461.

Wolhandler, Ruben, Arie Cattan, Ori Ernst, and Ido Dagan (2022). "How "Multi" is Multi-Document Summarization?" In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP 2022)*. Abu Dhabi, United Arab Emirates, pp. 5761–5769.

Wu, Chien-Sheng, Linqing Liu, Wenhao Liu, Pontus Stenetorp, and Caiming Xiong (2021). "Controllable Abstractive Dialogue Summarization with Sketch Supervision". In: *Findings of the Association for Computational Linguistics: (ACL-IJCNLP 2021)*. Online, pp. 5108–5122.

Wu, Shuangzhi, Dongdong Zhang, Nan Yang, Mu Li, and Ming Zhou (2017). "Sequence-to-Dependency Neural Machine Translation". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*. Vancouver, Canada, pp. 698–707.

Xiao, Wen, Iz Beltagy, Giuseppe Carenini, and Arman Cohan (2022). "PRIMERA: Pyramid-based Masked Sentence Pre-training for Multi-document Summarization". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)*. Dublin, Ireland, pp. 5245–5263.

Xu, Canwen, Jiaxin Pei, Hongtao Wu, Yiyu Liu, and Chenliang Li (2020a). "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, pp. 3586–3596.

Xu, Jiacheng, Shrey Desai, and Greg Durrett (2020). "Understanding Neural Abstractive Summarization Models via Uncertainty". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*. Online, pp. 6275–6281.

Xu, Runxin, 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 (2020b). "Xiaomingbot: A Multilingual Robot News Reporter". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (ACL 2020)*. Online, pp. 1–8.

Xu, Song, Haoran Li, Peng Yuan, Youzheng Wu, Xiaodong He, and Bowen Zhou (2020c). "Self-Attention Guided Copy Mechanism for Abstractive Summarization". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*. Online, pp. 1355–1362.

Yang, Min, Chengming Li, Fei Sun, Zhou Zhao, Ying Shen, and Chenglin Wu (2020). "Be Relevant, Non-Redundant, and Timely: Deep Reinforcement Learning for Real-Time Event Summarization". In: *Proceedings of The 34th AAAI Conference on Artificial Intelligence (AAAI 2020)*. New York, United States, pp. 9410–9417.

Yang, Min, 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, pp. 3110–3119.

Yang, Qian, Rebecca J Passonneau, and Gerard De Melo (2016). "PEAK: Pyramid Evaluation via Automated Knowledge Extraction". In: *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI 2016)*. Phoenix, Arizona, pp. 2673–2680.

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le (2019). "Xlnet: Generalized Autoregressive Pretraining for Language Understanding". In: *Proceedings of the 33th Annual Conference on Neural Information Processing System (NeurIPS 2019)*. Vancouver, Canada, pp. 5754–5764.

Yao, Kaichun, Libo Zhang, Tiejian Luo, and Yanjun Wu (2018). "Deep Reinforcement Learning for Extractive Document Summarization". In: vol. 284, pp. 52–62.

Yasunaga, Michihiro, Jungo Kasai, Rui Zhang, Alexander R Fabbri, Irene Li, Dan Friedman, and Dragomir R Radev (2019). "Scisummnet: A Large Annotated Corpus and Content-impact Models for Scientific Paper Summarization with Citation Networks". In: *Proceedings of the 33th AAAI Conference on Artificial Intelligence (AAAI 2019)*. Honolulu, United States, pp. 7386–7393.

Yasunaga, Michihiro, 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)*. Vancouver, Canada, pp. 452–462.

Yin, Wenpeng 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, pp. 1383–1389.

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

and Amr Ahmed (2020). "Big Bird: Transformers for Longer Sequences". In: *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS 2020)*. Online, pp. 17283–17297.

Zajic, David M, Bonnie J Dorr, and Jimmy Lin (2008). "Single-document and Multi-document Summarization Techniques for Email Threads Using Sentence Compression". In: vol. 44. 4, pp. 1600–1610.

Zhang, Jianmin, 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, pp. 381–390.

Zhang, Jingqing, Yao Zhao, Mohammad Saleh, and Peter J. Liu (2020a). "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization". In: *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*. Online, pp. 11328–11339.

Zhang, Quanshi, Ying Nian Wu, and Song-Chun Zhu (2018). "Interpretable Convolutional Neural Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018)*. Salt Lake City, United States, pp. 8827–8836.

Zhang, Shiyue, Asli Celikyilmaz, Jianfeng Gao, and Mohit Bansal (2021). "EmailSum: Abstractive Email Thread Summarization". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*. Online, pp. 6895–6909.

Zhang, Tianyi, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi (2020b). "BERTScore: Evaluating Text Generation with BERT". In: *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020)*. Addis Ababa, Ethiopia.

Zhang, Tianyi, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi (2020c). "BERTScore: Evaluating Text Generation with BERT". In: *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020)*. Addis Ababa, Ethiopia.

Zhang, Wei Emma, Quan Z. Sheng, Ahoud Abdulrahmn F. Alhazmi, and Chenliang Li (2020d). "Adversarial Attacks on Deep-learning Models in Natural Language Processing: A Survey". In: vol. 11. 3, 24:1–24:41.

Zhang, Wei Emma, Quan Z. Sheng, Adnan Mahmood, Dai Hoang Tran, Munazza Zaib, Salma Abdalla Hamad, Abdulwahab Aljubairy, Ahoud Abdulrahmn F. Alhazmi, Subhash Sagar, and Congbo Ma (2020e). "The 10 Research Topics in the Internet of Things". In: *Proceedings of 6th IEEE International Conference*

on Collaboration and Internet Computing (CIC 2020). Atlanta, United States, pp. 34–43.

Zhang, Yong, Meng Joo Er, Rui Zhao, and Mahardhika Pratama (2016). "Multiview Convolutional Neural Networks for Multidocument Extractive Summarization". In: vol. 47. 10, pp. 3230–3242.

Zhao, Chao, Tenghao Huang, Somnath Basu Roy Chowdhury, Muthu Kumar Chandrasekaran, Kathleen R. McKeown, and Snigdha Chaturvedi (2022). "Read Top News First: A Document Reordering Approach for Multi-Document News Summarization". In: *Findings of the Association for Computational Linguistics (ACL 2022 findings)*. Dublin, Ireland, pp. 613–621.

Zhao, Jinming, Ming Liu, Longxiang Gao, Yuan Jin, Lan Du, He Zhao, He Zhang, and Gholamreza Haffari (2020). "SummPip: Unsupervised Multi-Document Summarization with Sentence Graph Compression". In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020)*. Online, pp. 1949–1952.

Zhao, Mingjun, Shengli Yan, Bang Liu, Xinwang Zhong, Qian Hao, Haolan Chen, Di Niu, Bowei Long, and Weidong Guo (2021). "QBSUM: A Large-scale Query-based Document Summarization Dataset from real-world applications". In: vol. 66, p. 101166.

Zhao, Wei, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger (2019). "MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance". In: *Proceedings of the Conference on Empirical Methods in Natural Language and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*. Hong Kong, China, pp. 563–578.

Zheng, Xin, 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, pp. 3151–3160.

Zhong, Ming, 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, pp. 6197–6208.

Zhong, Ming, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir R.

Radev (2021). “QMSum: A New Benchmark for Query-based Multi-domain Meeting Summarization”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2021)*. Online, pp. 5905–5921.

Zhou, Bolei, 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, pp. 2921–2929.

Zhu, Chenguang, Yang Liu, Jie Mei, and Michael Zeng (2021). “MediaSum: A Large-scale Media Interview Dataset for Dialogue Summarization”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2021)*. Online, pp. 5927–5934.

Zhu, Chenguang, Ruochen Xu, Michael Zeng, and Xuedong Huang (2020). “A Hierarchical Network for Abstractive Meeting Summarization with Cross-Domain Pretraining”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings (EMNLP 2020)*. Online, pp. 194–203.

Zhu, Junnan, 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, pp. 4154–4164.

Zopf, Markus (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, pp. 1687–1696.