3M:Multi-document Summarization Considering Main and Minor Relationship

Anonymous ACL-IJCNLP submission

Abstract

The multi-document summary task is an important branch of the information aggregation task. Compared with the single-document summary, the input of multi-document summary is much longer and the logic is more complicated. This article proposes a hypothesis: taking the content of a document as the main body and the content of other documents as auxiliary information, a summary that combines all the information in the document collection can be generated. Based on this assumption, the multi-document summarization task can select one main document, and then combine the information of other documents for summary generation. This paper combines CopyTransformer and the Maximal Marginal Relevance (MMR) to design **M**ulti-document summarization considering **M**ain and **M**inor relationship model(3M). Empirical results on the Multi-News and DUC 2004 dataset show that the 3M brings substantial improvements over several strong baselines, manual evaluation shows that the generated abstract is fluent and can better express the content of the main document. In addition, by selecting different main documents, 3M can generate multiple abstracts with different styles for a set of documents.

1 Introduction

Generative text summarization is a research difficulty and hotspot in the field of natural language processing. Its main task is to refine and summarize the key information of the input documents set, so as to generate the main content that can summarize the source text. The multi-document summarization task studied in this paper has a wide range of application scenarios, such as news collection summary extraction (Fabbri et al., 2019), opinion summarization from online forums(YING and Jiang, 2015), and search engines (Zopf, 2018; Wang et al., 2020; Pasunuru et al., 2021). In recent years, with the rapid development of sequence models, the research on the single-document summarization model for simple input has been relatively complete(Cho et al., 2014; Narayan et al., 2018; Zhang et al., 2020), but for the multi-document summarization with more complex input, the encoder-decoder framework used in the traditional single-document summary model is difficult to apply.

Specifically, it is more difficult to construct a multi-document summary data set, and there are fewer high-quality datasets with sufficient data, so the effect of supervised model training is not ideal. The overall length of the multi-document summary is longer, and the model is difficult to pay attention to the really important information. Different from long documents, there might be multiple sentences with almost the same semantics in a multi-document collection, which brings the problem of content redundancy. In addition, when different documents discuss the same topic, the opposite point of view may be put forward. These above problems are the focus of the previous multi-document summarization task(Fabbri et al., 2019).

Processing the problem of multi-document summarization, we propose a new assumption: taking one document as the main body and the other documents as auxiliary information can generate a great summary that combines all the information in the document collection. If the assumption is true, the inrfomation of minor documents can be compressed to solve the problem of too long and too redundant input, and through the selection of the theme document, it is also possible to determine the viewpoint orientation of the summary when the viewpoints in multiple documents differ.

The model is designed with an encoder-decoder structure. The encoder and decoder are both stacked by multiple network layers with similar

structures. In the encoder, each network layer includes two sublayers, which are the multi-head self-attention mechanism sublayer and the fully connected feedforward sublayer. The output of each sub-layer is also connected to a residual connection network and a layer standardization network. In the decoder, each layer includes three sublayers, including two multi-head self-attention mechanism sublayers and a fully connected feedforward sublayer. In addition, when decoding, a dynamic maximum boundary correlation algorithm (MMR) is introduced. Whenever a sentence is generated on the decoder side, by calculating the MMR score of the sentence, the attention distribution can be adjusted.

By processing the standard multi-document summary dataset, a dataset that meets the requirements can be obtained – the document with the highest similarity to the standard summary is selected as the main document, and the other documents are selected as minor documents. After the dataset has been processed, this article has done a lot of experiments on the multi-document dataset, including automatic evaluation experiments, manual evaluation experiments, and ablation experiments. The experiment results show that 3M make great improvement compared to previous models.

The contributions of this article are as follows:

- Proposed a new solution for multi-document summarization. The summary is constructed around an document as the main document, which solves the problems of long input, excessive redundancy, and contradictory abstract content;
- 3M can choose different documents as the main document, so that the perspective of the summary has a certain direction;
- Proposed a new model architecture, combining the transformer model and the MMR model to obtain a more readable text summary.

2 Related Work

In recent years, the research of single-document summarization model has achieved many phased results(Li et al., 2018; Zhang et al., 2020; Hasan et al.). During this period, more and more researchers have turned their attention to the field of multi-document summarization.

The task of multi-document summarization is difficult to obtain when constructing the data set. In this case, the unsupervised generative model is a better solution. Chu and Liu (2019) generated summaries by training two recurrent autoencoders (Cho et al., 2014) on the Yelp and Amazon reviews datasets(McAuley et al., 2015), and constructed the loss function from two aspects. Zhang et al. (2018) applied a hierarchical single-document summarization model to a multi-document scenario to learn the vector representation of each document input and adjust the parameters of the model; Lebanoff et al. (2018) proposed pointer generator, which is based on the traditional two-way LSTM, adds a pointer mechanism and an overlay mechanism to solve the unknown word problem and the repeated word problem. They introduced a Maximal Marginal Relevance (MMR) model based on the pointer-generator, which is essentially an extractive summary algorithm that can comprehensively consider the relevance and redundancy of the summary.

Some researchers apply an extraction algorithm to simplify the input of the model. This operation can reduce content redundancy to a certain extent, and finally train a generative model for the simplified input to obtain the final Summary. Liu et al. (2018) established the data set Wikisum. In the process of generating summaries, they first used TF-IDF, TextRank, SumBasic and other relatively basic extraction algorithms to filter the source document set, and then passed a standard Two-way LSTM model (encoder-decoder architecture with attention mechanism) to generate the final summary.

There are also researchers who directly train generative models on the parallel multi-document summarization corpus. Fabbri et al. (2019) established the Multi-News data set, which is also one of the main data sets used in this article. They also used the pointer-generator network and integrated the MMR model into it. Zhou et al. construct a heterogeneous graph network for multi-document summarization, which allows rich cross-document information to be captured. Pang et al. (2021) build the English AgreeSum dataset based on English Wikipedia current events portal(WCEP), and provide abstractive summaries that represent information common and faithful to all input articles.

3 Preliminaries

3.1 Maximal Marginal Relevance

Maximal Marginal Relevance(MMR) was proposed by Carbonell and Goldstein (1998). MMR is used for single-document summarization tasks as an extractive summarization algorithm. The main idea is to calculate the MMR score for each sentence in the document, and extract the sentences with a higher MMR score as a summary. The MMR algorithm will comprehensively consider the degree of relevance of each sentence to the central idea of the entire document and the diversity of the summary itself. The MMR score can be calculated by equation 1:

$$\begin{aligned} MMR = \arg\max_{D_i \in R \setminus S} [\lambda Sim_1(D_i, Q) \\ -(1-\lambda) \max_{D_j \in S} Sim_2(D_i, D_j)] \end{aligned} \quad (1)$$

where R represents the set of all sentences; S represents the set of sentences chosen to be summary; Q indicates the center of the entire document thought; D_i means a candidate sentence; D_j means a sentence in the summary.

3.2 CopyTransformer

CopyTransformer(Gehrmann et al., 2018) is the Transformer architecture that incorporates the Pointer mechanism, which is mainly used to solve the problem of OOV words in the input. Compared with the ordinary Transformer architecture, its decoder part divides the generation of words into two modes: one is the copy mode, which is to copy a specific word from the source text as the current output; the other is the generation mode, which is directly from the source text. Select a word in the output vocabulary as the current output.

Set the parameter p_{gen} during decoding, which characterizes the probability that the model uses the generated mode:

$$p_{gen} = \sigma(w_{h^*}^T h_t^* + w_s^T s_t + w_x^T x_t + b_{ptr}) \quad (2)$$

where h_t^* represents the context vector calculated using the attention mechanism; s_t denotes the current hidden state of the decoder; x_t is the input word vector of the decoder; $w_{h^*}^T$, w_s^T, w_x^T and b_{ptr} are all learnable parameters. The probability distribution of the generated mode is similar to the ordinary sequence-to-sequence model, which is obtained by using the Softmax function on the output vocabulary; the probability distribution of the replication mode is equivalent to the attention distribution from the decoder to the encoder at the current time step:

$$P_{vocab} = Softmax(V'(v[s_t, h_t^*] + b) + b') \quad (3)$$

$$P_{copy} = \sum_{i:w_i=w} a_i^t \quad (4)$$

where a_i^t represent the attention score of the i-th word; V, V', b and b' are all learnable parameters. The final vocabulary is the union of the output vocabulary and the set of input text words, and the probability distribution is given by the equation 5:

$$P(w) = p_{gen} P_{vocab}(w) + (1 - p_{gen}) P_{copy}(w) \quad (5)$$

4 The Proposed Method

This section proposes the **M**ulti-document summarization considering **M**ain and **M**inor relationship model(3M). 3M divides the input into two parts: main document and minor documents, these two parts are processed by an enhanced CopyTransformer with low-level Transformer layers and high-level Transformer layers. In the low-level layers,we added sentence masked multi-head attention to get the embedding of each sentence, and a dynamic MMR model is also added to adjust the attention distribution, thereby affecting the output of the final decoder. The specific structure is shown in Figure 1.

4.1 Low-level Transformer Layer

It can be seen from Figure 1 that the low-level Transformer layer in the decoder is exactly the same as the original Transformer layer(Vaswani et al., 2017). In the encoder, the low-level Transformer layer is used to learn the contextual connections between words in the input sequence, the multi-head attention sublayer of the encoder is divided into two modules, and these two modules use two masking mechanisms–word mask and sentence mask. The main function of the sentence mask is to prevent the semantic crossing between sentences, and only let the model learn the contextual semantics of each word in its sentence. Since 3M introduced a dynamic MMR model to the Transformer architecture, and the MMR algorithm uses

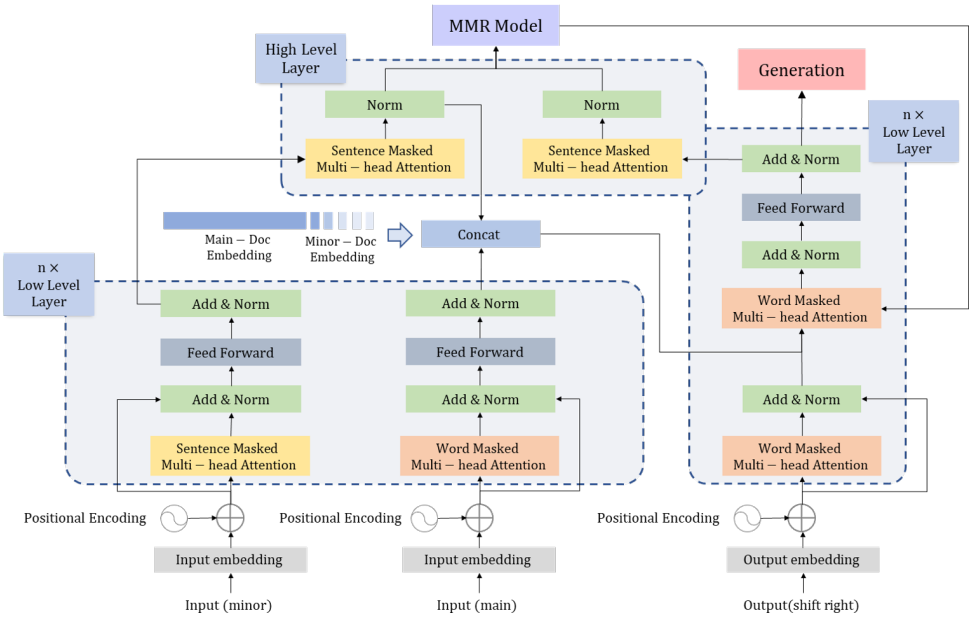

Figure 1: Overall architecture of the model 3M

sentences as the basic unit of MMR scores, a sentence mask is designed here to obtain an accurate sentence encoding, which is then input to the dynamic MMR model. In addition, in order to reduce the distraction caused by long input, the encoder uses sentence embedding to summarize the content of the minor documents, which reduces the output scale of the encoder.

$\{t_1, t_2, \ldots, t_m\}$ is the word sequence of the input, we use x_i and y_i to represent the output of the word t_i under the word mask and sentence mask respectively, and let $X = [x_1; x_2; \ldots; x_m]$, $Y = [y_1; y_2; \ldots; y_m]$. The output of the low-level Transformer layer under the word mask is consistent with the original Transformer; the sentence mask sets the attention value of all words outside the sentence to negative infinity, so the output of the low-level Transformer layer only contains the contextual information of the word in its corresponding sentence. X is used to calculate the key vector to get the attention distribution from encoder to decoder:

$$Q_k = XW_k^Q \tag{6}$$

$$K_k = XW_k^K \tag{7}$$

$$a_k^w = Softmax(\frac{Q_k K_k^T}{\sqrt{d_{head}}}) \tag{8}$$

where $W_k^Q \in R^{d \times d_{head}}$ and $W_k^K \in R^{d \times d_{head}}$ is learnable linear mapping matrix; $k \in \{1, 2, \ldots, h\}$ represents the k-th Transformer head; d represents input and output dimension of the each sub-layer in the 3M model; d_{head} represents the dimension of Transformer head; a_k^w means the attention distribution.

In particular, in low-level Transformer layer, we additionally encodes the word sequence of the main document at the word level, and the output X_{main} will be used as part of the encoder output.

4.2 High-level Transformer Layer

3M adds a high-level Transformer layer on the top of the low-level Transformer layer on the encoder and decoder sides. Intuitively, the sentence embedding should be calculated from the output of the sentence-masked multi-head attention corresponding to all words in the sentence, and the algorithm needs to reduce the dimensionality of the vector. Specifically, for the sentence s_i, its sentence embedding u_i should be calculated by $y_j, y_{j+1}, \ldots, y_{j+l_i}$, where l_i represents the length of the i-th sentence.

The high-level Transformer layer introduces a two-factor multi-head attention sublayer. The traditional multi-head attention sublayer involves the calculation of three factors–queries, keys, and values. In contrast, the two-factor multi-head attention sublayer only calculates two factors–the self-attention value scores and the values:

$$S_k = Y_{s_i} W_k^S \tag{9}$$

$$V_k = Y_{s_i} W_k^V \tag{10}$$

the encoder and decoder are the same in the high-level Transformer structure, take the encoder as an example, $Y_{s_i} = [y_j; y_{j+1}; \ldots; y_{j+l_i}]$ represents the matrix of the input sentence; $W_k^S \in R^{d\times 1}$and $W_k^V \in R^{d\times d_{head}}$ is learnable matrices. The self-attention value scores S_k is subjected to the Softmax operation to obtain the self-attention distribution of each word in the sentence s_i:

$$a_k^S = Softmax(S_k) \tag{11}$$

Then the self-attention distribution vector a_k^S is weighted and summed with the values vector to get the context vector representing the sentence s_i in the k-th semantic subspace (Transformer head):

$$c_i^k = a_k^S V_k \tag{12}$$

$$u_i = LN(W_c[c_i^1; c_i^2; \ldots; c_i^h]) \tag{13}$$

The sentence mask mechanism is also used in the dual-factor multi-head self-attention sublayer, thus the vector representation of each sentence is only related to the output of all words in the sentence at the low-level Transformer layer.

In particular, in the low-level layer of decoder, the input is the word-level encoding of the main document X_{main} and the sentence-level encoding of all documents except the main document $U_{\backslash main}$ spliced together.

4.3 Dynamic MMR Model

The dynamic MMR model takes all sentence representations and summary representations as input, and calculates the MMR score for each sentence s_i in the input sequence.

In realization, dynamic MMR model is modified on the basis of equation 1, it uses the source sentence embedding u_i to represent D_i, uses summary representation v_{sum} to replace Q, and uses current decoded target sentence's embedding v_j to represent D_j. Therefore, equation 1 can be rewritten as:

$$\begin{aligned} MMR_i = {} & \lambda Sim_{i1}(u_i, v_{sum}) \\ & -(1-\lambda)\max_j Sim_2(u_i, v_j) \end{aligned} \tag{14}$$

$$v_{sum} = W_Z Z + b_Z \tag{15}$$

$$Sim_1(u_i, v_{sum}) = \sigma(u_i^T W_{sim1} v_{sum}) \tag{16}$$

$$Sim_2 = \max_j \frac{exp(sim_{ij})}{\sum_j exp(sim_{ij})} \tag{17}$$

$$Sim_{ij} = w_{sim}^T tanh(W_u u_i + W_v v_j + b_{attn}) \tag{18}$$

$W_{sim1}, W_Z, b_z, w_{sim}^T, W_u, W_v, b_{attn}$ are learnable matrices, and λ is an artificial experience value, we set $\lambda = 0.5$ according to Liu et al. (2020). We uses a bilinear function to determine Sim_1, the input v_{sum} is calculated by the output matrix of the last layer of the lower-order transformer on the decoder side. For the definition of Sim_2, we calculate the similarity value of the candidate sentence s_i with multi-layer perceptron algorithm, and then use the Softmax function to convert all the similarity values into a probability distribution.

Taking into account that in the encoding process, the word level information of the main document and the sentence level information of the minor documents are spliced, so the attention of the sentence vector needs to be recalculated. Here we combine the MMR score to calculate the attention represented by the sentence distribution. The MMR score can guide the decoder to comprehensively consider the degree of correlation between the output sentence and the original document and the redundancy of the generated sentence when generating the summary, while the MMR score is obtained by subtracting two positive terms, we need to set it to a non-negative value for easy calculation, so we make the following processing:

$$MMR_i' = \frac{exp(MMR_i)}{\sum_i exp(MMR_i)} \tag{19}$$

$$a_{sen_i} = Mean_a MMR_i' \tag{20}$$

a_{sen_i} represents the attention of i-th sentence, $Mean_a$ is the mean value of the attention of the words in the minor document.

5 Experiments

We evaluate our model on two major datasets used in the literature of multi-document summarization – Multi-News (Fabbri et al., 2019) and DUC 2004 datasets.

Partition	Multi-News			DUC-2004		
	R-1	R-2	R-SU4	R-1	R-2	R-SU4
ext-LexRank	38.27	12.70	13.20	28.90	5.33	8.76
ext-TextRank	38.44	13.10	13.50	33.16	6.13	10.16
ext-MMR	38.77	11.98	12.91	30.14	4.55	8.16
abs-Pointer-Gen	41.85	12.91	16.46	31.43	6.03	10.01
abs-PG-MMR	40.55	12.36	15.87	36.42	9.36	**13.23**
abs-CopyTransformer	43.57	14.03	17.37	28.54	6.38	7.22
abs-Hi-MAP	43.47	14.89	17.41	35.78	8.90	11.43
abs-3M(Our Model)	**45.34**	**16.20**	**19.15**	**37.35**	**9.60**	12.29

Table 1: ROUGE F_1 scores on Multi-News and DUC 2004 datasets

The Multi-News dataset was proposed by Fabbri et al. (2019), consists of news articles and human-written summaries. The dataset comes from a diverse set of news sources, and contains 44972 instances for training, 5622 for validation, and 5622 for inference. DUC 2004 is a standard multi-document summarization test set, which contains only 50 document clusters. We treat it as an additional test set.

We use tf-idf (Ramos et al., 2003) to calculate the text similarity scores of all reference documents and gold summary, and set the document with the highest similarity score as the main document. The input of the model is a mega-document composed of multiple documents, the upper limit of the input length L is 1000, which is a suitable value obtained through experiments, and the extra part will be cropped.

3M contains 4 low-level Transformer layers and a high-level Transformer layer. We train our model for 40000 steps using Adam (Kingma and Ba, 2014) with learning rate of 0.7, $\beta_1 = 0.9, \beta_2 = 0.998$. We apply dropout with a rate of 0.2 and label smoothing of value 0.1. The model dimension d is 512, the number of heads is h is 8 and the feed-forward hidden size d_f is 2048. In the process of generating abstracts, we introduced beam search and coverage mechanisms (Gehrmann et al., 2018) in the generator to ensure that the generated abstracts have low redundancy and sufficient readability.

In addition to using ROUGE scores to evaluate the accuracy of the generated summaries, we also recruited 5 volunteers to evaluate the ability to generate summaries of 3M.

5.1 Baselines

We compare our model 3M with the following extractive and abstractive summarization methods.

LexRank & TextRank(Erkan and Radev, 2004; Mihalcea and Tarau, 2004) are two graph-based ranking methods that can be used for extractive summarization.

MMR (Carbonell and Goldstein, 1998) is a method combining query-relevance with information-novelty to extract important sentences.

Pointer-Gen is a generative summary model proposed by See et al. (2017), which is based on a Bi-LSTM structure and introduces a unique pointer mechanism and coverage mechanism.

PG-MMR is the adapted pointer-generator model introduced by Lebanoff et al. (2018), which mutes sentences that receive low MMR scores.

CopyTransformer is the generative summary model proposed by Gehrmann et al. (2018). The CopyTransformer model is the application of the pointer mechanism on the Transformer architecture. Following the example of Gehrmann et al. (2018), this paper uses a 4-layer network structure.

Hi-MAP (Fabbri et al., 2019) extends the PG network into a hierarchical network, and it also use the MMR scores of the sentences to improve the performance of the decoder.

5.2 Results

Automatic evaluation experiment

Table 1 lists the evaluation results of different models in the Multi-News and DUC 2004 datasets. Among them, ext means that the model is an extractive model, and abs means that the model is a generative model.

Compare to the baseline models, our 3M model yields much better results as shown in Table 1. On the Multi-News dataset, result shows that 3M

Model	Grammar	Referential	Clarity	Focus	Structure&Coherence
PG-MMR	-0.14	-0.07	-0.02	-0.21	-0.19
CopyTransformer	0.03	-0.04	-0.06	0.03	-0.01
Hi-MAP	0.03	-0.03	-0.07	0.07	0.00
3M	0.08	0.14	0.15	0.11	0.20

Table 2: Results of human evaluation on five metrics

achieves the best performance, and outperforms Hi-MAP by (+1.87 ROUGE-1, +1.31 ROUGE-2, +1.74 ROUGE-SU4) points. On the DUC 2004, 3M gets the highest points on ROUGE-1 and ROUGE-2, which outperforms Hi-MAP by (+1.57 ROUGE-1, +0.70 ROUGE-2) points, while model PG-MMR has the highest ROUGE-SU4 point.

Manual evaluation experiment

We selected five volunteers to conduct two manual evaluations to test the quality of the summaries generated by the 3M and whether the 3M model pays more attention to the information of the main document.

In the first experiment, 20 document sets were selected, and four models(PG-MMR, CopyTransformer, Hi-MAP, 3M) were used to generate corresponding summaries. Volunteers were asked to evaluate the quality of summaries from five aspects including grammar, non-redundancy, referential clarity, focus and structure&coherence. In the scoring strategy, the same Best-Worst Scaling method as Fabbri et al. (2019) is adopted. For each evaluation index, the score S of each model is equal to C_{best} (the number of times the model is selected as the best) minus C_{worst} (the number of times the model is selected as the worst), and then divided by c_{total} (the total number of comparisons):

$$S = \frac{C_{best} - C_{worst}}{C_{total}} \tag{21}$$

From Table 2 we can see the results of human evaluation on five metrics. Our model 3M is superior to the three models for comparison in every indicator, especially in terms of referential clarity and structure&coherence. Compared with other models, 3M mainly refers to one document, so it usually has more advantages in correspondence and article structure. And with the dynamic MMR model, 3M can effectively consider relevance and redundancy jointly.

In the second set of experiments, we selected 40 document sets, randomly selected one document in the set as the main document, and generated corresponding summary with 3M. Volunteers need to find the corresponding main document in the document set through the summaries. Finally, the accuracy of the prediction results is 92.0%, which means that 3M does use the main document as the largest reference for generating summaries.

Ablation experiment

Based on the Transformer architecture, 3M has added multiple mechanisms to improve the performance of the model. We have verified the effectiveness of these mechanisms. The ablation experiment used the ROUGE score as an evaluation index, and was verified under the Multi-News and DUC 2004 data set.

Table 3 shows the results on the Multi-News and DUC 2004 data set. Compared models include 3M and its variants with static MMR scores(Static MMR), without minor documents(without MD), without discrimination between main and minor documents(without discrimination), and randomly choosing the main document(Random Main). 3M(static MMR) compute static MMR scores only at the end of the decoder. 3M(without MD) masked all the output of the encoder corresponding to minor documents, only summarizes the main document. 3M(without discrimination) treats the main document and the minor documents equally, doesn't use sentence embeding to abstract minor documents, which is similar to Liu et al. (2020). 3M(Random Main) chooses main document randomly, and also sorts the minor documents randomly.

It is not difficult to see from the results that the dynamic MMR model has greatly improved the quality of the generated summaries. From the result of 3M(without MD), we can see that our 3M model not only considers the main document, but also give a thought to the supplementary role of minor documents. Comparing the results of the 3M(without discrimination) group, we can know that it is meaningful for us to take a simplified representation to the token of the minor documents and generate a shorter encoder output. The experi-

Partition	Multi-News			DUC-2004		
	R-1	R-2	R-SU4	R-1	R-2	R-SU4
3M(Static MMR)	43.71	14.63	17.66	35.97	9.00	11.43
3M(without MD)	44.15	15.31	18.27	36.55	9.19	11.90
3M(without Discrimination)	44.77	15.53	18.46	37.14	9.45	12.12
3M(Random Main)	43.02	13.90	17.44	35.42	8.88	11.27
3M	**45.34**	**16.20**	**19.15**	**37.35**	**9.60**	**12.29**

Table 3: Results of ablation experiments on dataset Multi-News and DUC-2004.

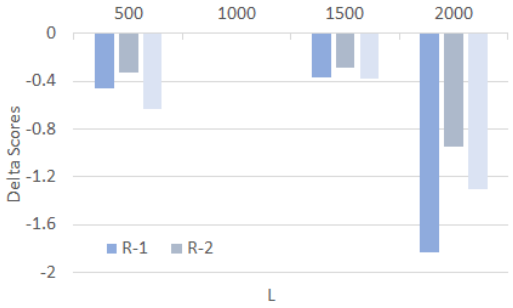

Figure 2: Delta ROUGE scores under Multi-News dataset when L=500,1000,1500,2000.

ment of the 3M (Random Main) randomly selected the main document, so it did not focus on the document most relevant to the gold summary. When using our model to generate the summary, the main documents would omit some important information instead, so the scores are relatively low. It's worth noting that the 3M model used 3M (Random Main) or 3M (without Discrimination) in the face of multi-document summarization tasks without specifying the main document. The former is more suitable for tasks with more similar content in multiple documents, it performs worse when there are conflicting views between different documents; the latter is suitable when the overall length of multiple documents is small, otherwise it is easy to omit the key information.

Input length setting experiment

Liu and Lapata (2019) sets the input length $L = 500$ in a similar multi-document summarization task. Taking into account the compression processing of the input in the encoder of our model 3M, the representation unit of the minor documents is one sentence, so the input length L can be set larger. In the case of ensuring that the input information is not omitted, the model's attention will not be distracted, and the generated summary can also focus on the more important parts.

We experimented with the input length L during training, and L was set to 5000, 1000, 1500, 2000. We set the ROUGE score when $L = 1000$ as the reference value, and calculate delta scores according to the reference value. From Figure 2 we can see that 3M get the best ROUGE scores when $L = 1000$. When L is set to 500, the number of input tokens is too small, and even in some cases, the length of a single document will exceed 500, and a lot of input information is deleted, so the score obtained is relatively low. When L is set to above 1500, the too long input brings too much irrelevant information, which will have a certain impact on the redundancy and focus of the generated summary.

6 Conclusion

In this article, for the problems of long input, excessive redundancy, and contradictory content in the multi-document summarization task, we put forward a hypothesis – by choosing one document as the main document, and other documents as minor documents, high-quality summarization can also be generated. On the basis of this assumption, we proposed a 3M model, which is based on the CopyTransformer model and adds a dynamic MMR mechanism. Experimental results demonstrates that our 3M model made considerable progress compared to several strong baselines, which also proves that our hypothesis is reasonable.

References

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder

for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Eric Chu and Peter Liu. 2019. Meansum: a neural model for unsupervised multi-document abstractive summarization. In *International Conference on Machine Learning*, pages 1223–1232. PMLR.

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

Alexander R Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. *arXiv preprint arXiv:1906.01749*.

Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109.

Tahmid Hasan, Abhik Bhattacharjee, Md Saiful Islam, Kazi Samin, Yuan-Fang Li, Yong-Bin Kang, M Sohel Rahman, and Rifat Shahriyar. Xl-sum: Large-scale multilingual abstractive summarization for 44 languages.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Logan Lebanoff, Kaiqiang Song, and Fei Liu. 2018. Adapting the neural encoder-decoder framework from single to multi-document summarization. *arXiv preprint arXiv:1808.06218*.

Wei Li, Xinyan Xiao, Yajuan Lyu, and Yuanzhuo Wang. 2018. Improving neural abstractive document summarization with explicit information selection modeling. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 1787–1796.

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

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

Yiding Liu, Xiaoning Fan, Jie Zhou, Chenglong He, and Gongshen Liu. 2020. Learning to consider relevance and redundancy dynamically for abstractive multi-document summarization. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 482–493. Springer.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 43–52.

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

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

Richard Yuanzhe Pang, Adam D Lelkes, Vinh Q Tran, and Cong Yu. 2021. Agreesum: Agreement-oriented multi-document summarization.

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

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Kexiang Wang, Baobao Chang, and Zhifang Sui. 2020. A spectral method for unsupervised multi-document summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 435–445.

DING YING and Jing Jiang. 2015. Towards opinion summarization from online forums. ACL.

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

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

Hao Zhou, Weidong Ren, Gongshen Liu, Bo Su, and Wei Lu. Entity-aware abstractive multi-document summarization.

Markus Zopf. 2018. Auto-hmds: Automatic construction of a large heterogeneous multilingual multi-document summarization corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.