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

Anonymous ACL submission

Abstract

The multi-document summarization (MDS) is an important branch of information aggregation. Compared with the single-document summary (SDS), MDS has three major challenges: (1) MDS involves too large search space to capture the attention; (2) the input of MDS contains a lot of redundant information and more complex logical relationships; (3) the different opinions of documents bring contradictions. To complete these three main challenges, we combine the Transformer and the Maximal Marginal Relevance (MMR) to design **M**ulti-document summarization considering **M**ain and **M**inor relationship (3M) model. In this model, we take one document as the main body and use the information of other documents as an addition to modifying the generation of the summary. Therefore, we can reduce the search space and ignore the redundancy in the minor documents. Empirical results on the Multi-News and DUC 2004 dataset show that the 3M brings substantial improvements over several strong baselines, the 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 one set of documents.

1 Introduction

The multi-document summary (MDS) is a research challenge and hotspot in the field of natural language processing. Its main task is generating a short and informative summary across a set of topic-related documents. MDS has a wide range of application scenarios, such as news collection summary extraction (Fabbri et al., 2019b), 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 (SDS) model for simple input has been well studied (Cho et al., 2014; Narayan et al., 2018; Zhang et al., 2020), but for MDS, the traditional encoder-decoder framework used in the SDS is difficult to apply, and MDS faces many challenges:

- The input of MDS are multiple documents, many works concatenate multiple documents into a flat sequence(Fabbri et al., 2019a; Liu and Lapata, 2019a), so the overall input is always longer than SDS, and the search space is also larger (Cohan et al., 2018);
- There might be multiple sentences with almost the same semantics in a multi-document collection, which brings the problem of content redundancy(Fabbri et al., 2019a);
- Different input documents may not have the same opinions on the same issue, and there may be contradictions during summary generation.

To overcome these challenges, we propose a new way for MDS: taking one document as the main body and the other documents as auxiliary information. Through the selection of the main document, the information of minor documents can be compressed to solve the problem of long and redundant input, and it is also possible to determine the viewpoint of the summary when it differs in multiple documents.

3M model is designed with an encoder-decoder structure. The encoder and decoder are both stacked by multiple network layers with similar structures, and we added a pointer mechanism on the decoder side according to Gehrmann et al. (2018). In the encoder, we encode the input of the main document and the minor document separately to obtain vector representations with different granularities, and send them to the decoder after splicing. In addition, Maximal Marginal Relevance

(MMR) (Carbonell and Goldstein, 1998) is introduced during decoding. We dynamically calculate the MMR score, whenever a sentence is generated on the decoder side, we adjust the input attention distribution according to the MMR score.

During training, we process the standard multi-document summary dataset – 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. During testing and verifying, we adopt two methods to select the main document: (1) obtain the document most closely related to other documents through an algorithm based on TextRank (Mihalcea and Tarau, 2004); (2) directly specify the main document manually. In such settings, we have 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 makes 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 a document as the main document, which solves the problems of large search space and excessive redundancy;
- 3M can choose different documents as the main document, so that the perspective of the summary has a certain attitude;
- 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, great progress has been made in the study of SDS (Paulus et al., 2017; Li et al., 2018; Gehrmann et al., 2018; Narayan et al., 2018; Perez-Beltrachini et al., 2019; Sharma et al., 2019; Zhang et al., 2020; Hasan et al.; Arakawa and Yakura, 2021). During this period, more and more researchers have turned their attention to the field of MDS.

The task of MDS is difficult to obtain in datasets construction. In this case, the unsupervised generative model is a good solution. Chu and Liu (2019) generated summaries by training two recurrent autoencoders on the Yelp and Amazon reviews datasets(McAuley et al., 2015), and constructed the loss function from two aspects. Another way to bypass this problem is model adaptation. 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; Lebanoff et al. (2018) proposed pointer generator, which adds a pointer mechanism and an overlay mechanism to solve the unknown word problem and the repeated word problem. Liu and Lapata (2019b) introduced a MMR model based on the pointer-generator, which is essentially a 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) 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. Zhong et al. (2020) compares the evaluation methods of Sentence-Level and Summary-Level, and proposes a summary method based on matching on this basis to extract the summary.

There are also researchers who directly train abstractive models on the parallel MDS corpus. Fabbri et al. (2019b) 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. Fan et al. (2019) further propose to construct a local knowledge graph from documents and then linearize the graph into a sequence to better sale sequence-to-sequence models to multi-document inputs. Zhou et al. (2021a) build 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 SDS task as an extractive summarization algorithm. 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:

$$\begin{aligned}\text{MMR} = \arg\max_{D_i \in R \setminus S}[\lambda \text{Sim}_1(D_i, Q) \\ -(1-\lambda)\max_{D_j \in S}\text{Sim}_2(D_i, D_j)],\end{aligned} \tag{1}$$

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 (Vinyals et al., 2015), 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 selecting a word in the output vocabulary.

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

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

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 at the current time step:

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

$$P_{copy} = \sum_{i:w_i=w} a_i^t, \tag{4}$$

where a_i^t represents 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 equation 5:

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

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 add sentence masked multi-head attention to get the embedding of each sentence. A dynamic MMR model is 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 according to (Yang et al., 2019). 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 sentences as the basic unit of MMR scores, a sentence mask is designed here to obtain an accurate sentence encoding. In addition, in order to reduce the distraction caused by long input, we adopted a more coarse-grained expression for the vector representation of minor documents — the encoder uses sentence encoding 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. And we use x_i, y_i to represent the output of

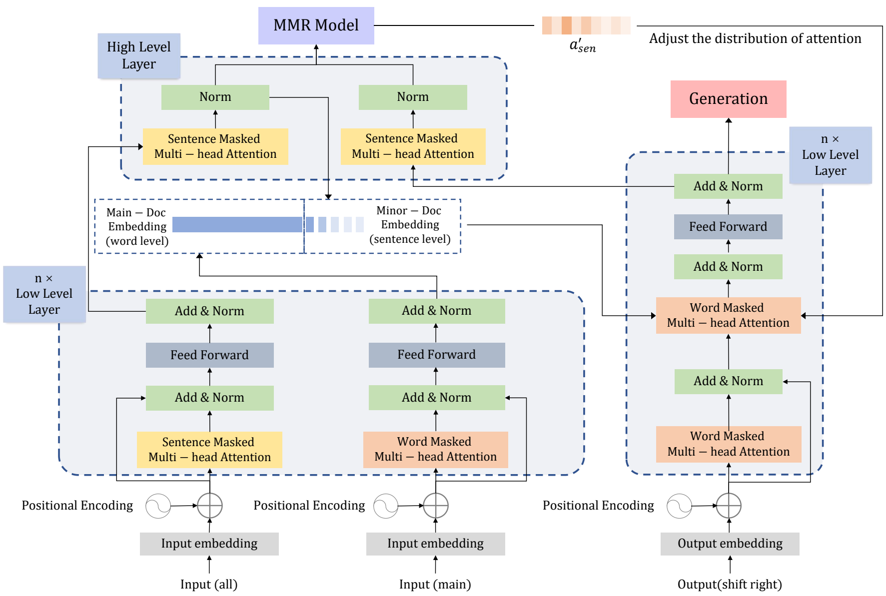

Figure 1: Overall architecture of the model 3M.

i-th word under word masked attention and sentence masked attention. $X = [x_1; x_2; \ldots; x_m]$ is used to calculate the attention distribution:

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

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

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

where $W_k^Q \in \mathbb{R}^{d \times d_{head}}$ and $W_k^K \in \mathbb{R}^{d \times d_{head}}$ are learnable 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 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. The sentence encoding 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 encoding u_i should be calculated from $Y_{s_i} = [y_j; y_{j+1}; \ldots; y_{j+l_i}]$, where l_i represents the length of the i-th sentence.

The encoder and decoder are similar in the high-level Transformer structure, take the encoder as an example. 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 — self-attention scores and values:

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

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

$W_k^S \in \mathbb{R}^{d \times 1}$and $W_k^V \in \mathbb{R}^{d \times d_{head}}$ are 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:

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

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):

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

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

The sentence mask mechanism is also used in the dual-factor multi-head self-attention sublayer. In particular, the input of the decoder is the word-level embedding of the main document X_{main} and the sentence-level embedding of all documents except the main document $U_{\setminus main}$.

4.3 Dynamic MMR Model

The dynamic MMR model takes sentence embedding and summary representation as input, and calculates the MMR score for each sentence s_i.

In realization, dynamic MMR model is modified on the basis of equation 1, it uses the source sentence encoding u_i to represent D_i, uses decoded 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:

$$\begin{aligned} \mathrm{MMR}_i &= \lambda \mathrm{Sim}_1(u_i, v_{sum}) \\ &-(1-\lambda)\max_j \mathrm{Sim}_2(u_i, v_j), \end{aligned} \tag{14}$$

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

$$\mathrm{Sim}_1(u_i, v_{sum}) = \sigma(u_i^T W_{sim1} v_{sum}), \tag{16}$$

$$\mathrm{Sim}_2 = \max_j \frac{\exp(sim_{ij})}{\sum_j \exp(sim_{ij})}, \tag{17}$$

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

$W_{sim1}, W_Z, b_z, w_{sim}^T, W_u, W_v, b_{attn}$ are model parameters, Z is the embedding of the decoded sentences from decoder, and λ is an artificial experience value, we set $\lambda = 0.5$ according to Liu et al. (2020). We use 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 concatenated, so the attention of the sentence vector needs to be recalculated during decoding. 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, 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:

$$\mathrm{MMR}'_i = \frac{\exp(\mathrm{MMR})}{\sum_j \exp(\mathrm{MMR})}, \tag{19}$$

$$a'_{sen_i} = \frac{a_{sen_i}\mathrm{MMR}'_i}{\sum_j a_{sen_j}\mathrm{MMR}'_i}. \tag{20}$$

a_{sen_i} represents the attention of i-th sentence of minor documents, and we adjust a_{sen_i} with the score MMR'.

5 Experiments

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

The Multi-News dataset was proposed by Fabbri et al. (2019b), consisting 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.

During training, we calculate ROUGE scores of all input documents with the gold summary as the text similarity scores. In terms of the specific implementation, we take the mean of ROUGE-1, ROUGE-2 and ROUGE-SU4 scores as the similarity score. Then we 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 1200, which is a suitable value obtained through experiments, and the extra part will be cropped. During validation and testing, we adopt two methods to select the main document:

- 3M(Gold) Calculate the text similarity scores (same) between all input documents and the gold summary, and directly select the document with the highest score as the main document. This method introduces additional information.

- 3M(TextRank) Obtain the main document through a algorithm based on TextRank (Mihalcea and Tarau, 2004). We take each input sentence as a node. The similarity scores of all nodes are calculated by the TextRank, and finally the sum of the scores of nodes in one document is used as the document importance score. We choose the document with the highest score as the main document.

3M contains 4 low-level Transformer layers and 1 high-level Transformer layer. We train our model for 40000 steps using Adam (Kingma and Ba, 2014) with a learning rate of 0.7. 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. With the above settings, it takes about 120 hours to complete the experiment on a single 1080Ti.

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 introduces pointer 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).
Hi-MAP (Fabbri et al., 2019b) extends the PG network into a hierarchical network, and it also uses the MMR to improve the performance of the decoder.
GraphSum is proposed by Li et al. (2020), which introduced graph information to adjust the attention distribution during encoding and decoding.
EMSum (Zhou et al., 2021b) proposed a framework with a graph consisting of text units and entities.

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. 3M(TextRank) means that we use TextRank to get the main document, while 3M(Gold) means that we directly select the document with the highest similarity score as the main document.

Compare to the baseline models, our 3M model yields much better results as shown in Table 1. On the Multi-News dataset, results show that 3M(Gold) achieves the best performance on R-1, R-2 and R-SU4(achieves +1.40/+0.02/+5.60 improvements compared with the second one), while EMSum performs best on R-L, the gap between EMSum and 3M is 2.68; 3M(TextRank) also performs better than others on R-1 and R-SU4, but relatively low on R-2 and R-L compared with EMSum. On the DUC-2004 dataset, our 3M(TextRank) and 3M(Gold) performs better than other models(the gold achieves +2.83/+1.34/+0.13 improvements compared with the second one). Considering all these metrics together, the results show the effectiveness of our model.

It is worth noting that the scores of 3M(TextRank) are relatively low compared to 3M(Gold). This is because when 3M(TextRank) selects the main document, it may not necessarily select the document that is closest to the gold summary. 3M(TextRank)'s performance is comparable to GraphSum, which proves that taking one document as the main document to generate summary is a feasible method, and the outstanding performance of 3M(gold) proves that we can obtain excellent summaries in the task of multi-document summarization with the designated main document.

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-GraphSum	45.02	16.69	-	-	-	-
abs-EMSum	45.57	17.71	-	-	-	-
abs-3M(TextRank)	46.05	16.75	22.39	38.24	10.01	12.91
abs-3M(Gold)	**46.97**	**17.73**	**23.01**	**39.25**	**10.70**	**13.36**

Table 1: ROUGE F_1 scores on Multi-News and DUC 2004 datasets. The results of GraphSum and EMSum are token from (Zhou et al., 2021b). The evaluation indicators of these two models are F_1 scores of ROUGE-1, ROUGE-2 and ROUGE-L. On the dataset Multi-News, the ROUGE-L socres of GraphSum, EMSum, 3M(TextRank) and 3M(Gold) are 22.50, **26.43**, 22.47 and 23.75.

Model	Grammar	Referential	Clarity	Focus	Structure&Coherence
PG-MMR	-0.100	-0.055	0.015	-0.160	-0.055
CopyTransformer	**0.045**	-0.050	-0.040	0.030	0.005
Hi-MAP	0.010	-0.030	0.000	0.060	0.005
3M	**0.045**	**0.135**	**0.025**	**0.070**	**0.045**

Table 2: Results of human evaluation on five metrics

Manual evaluation experiment

We selected five volunteers to evaluate the quality of the summaries generated by the 3M. 40 document sets were selected, and four models (PG-MMR, CopyTransformer, Hi-MAP, 3M) were used to generate 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. (2019b) is adopted. For each evaluation indicator, 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).

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.

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 to evaluate the performance of the model, and the experiments were verified under the Multi-News and DUC 2004 data set.

Table 3 shows the ablation experiment 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 embedding to abstract minor documents, which is similar to Liu et al. (2020). 3M(Random Main)

Partition	Multi-News			DUC-2004		
	R-1	R-2	R-SU4	R-1	R-2	R-SU4
3M(Static MMR)	44.17	15.00	17.78	36.17	9.12	11.76
3M(without MD)	45.01	15.47	18.37	36.96	9.43	12.11
3M(without Discrimination)	44.99	15.61	18.84	37.13	9.47	12.22
3M(Random Main)	44.58	14.41	17.98	36.33	9.12	11.43
3M	**46.97**	**17.45**	**23.01**	**39.25**	**10.70**	**13.36**

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

chooses the main document randomly, and also sorts the minor documents randomly.

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 information 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 experiment of the 3M (Random Main) randomly selected the main document, so it did not focus on the document most relevant to the gold summary, and the score is 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

Taking into account the compression processing of the input in the encoder of 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 600, 1200, 1800 and 2400. We set the ROUGE scores when =1200 as the reference value, and calculate delta scores according to the reference value. From Figure 2 we can see that 3M gets the best ROUGE scores when L=1200. When L is set to 600, the number of input tokens is too small, and even in some cases, the length of a single document will exceed 600, and a lot of input information is deleted, so the score obtained is lower. When L is set to above 1800, the too-long input brings too much irrelevant information, which will have a certain impact on the redundancy and focus of the generated summary.

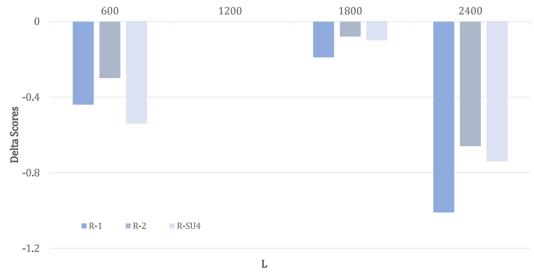

Figure 2: Delta ROUGE scores under Multi-News dataset when L=600, 1200, 1800, 2400.

6 Conclusion

In this article, for the problems of large search space, excessive redundancy, and contradictory content in the multi-document summarization task, we choose one document as the main document, and other documents as minor documents. On this basis, we proposed a 3M model, which is based on the CopyTransformer and adds a dynamic MMR mechanism. Experimental results demonstrate that our 3M model makes considerable progress compared to several strong baselines, which proves that our method considering main and minor relationship is reasonable.

References

Riku Arakawa and Hiromu Yakura. 2021. Reaction or speculation: Building computational support for users in catching-up series based on an emerging media consumption phenomenon. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1):1–28.

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.

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

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

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

Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2019. Using local knowledge graph construction to scale seq2seq models to multi-document inputs. *arXiv preprint arXiv:1910.08435*.

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, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du. 2020. Leveraging graph to improve abstractive multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6232–6243.

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

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

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.

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

Laura Perez-Beltrachini, Yang Liu, and Mirella Lapata. 2019. Generating summaries with topic templates and structured convolutional decoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5107–5116.

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

Eva Sharma, Luyang Huang, Zhe Hu, and Lu Wang. 2019. An entity-driven framework for abstractive summarization. *arXiv preprint arXiv:1909.02059*.

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.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *arXiv preprint arXiv:1506.03134*.

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.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

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.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuan-Jing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208.

Hao Zhou, Weidong Ren, Gongshen Liu, Bo Su, and Wei Lu. 2021a. Entity-aware abstractive multi-document summarization.

Hao Zhou, Weidong Ren, Gongshen Liu, Bo Su, and Wei Lu. 2021b. Entity-aware abstractive multi-document summarization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 351–362.

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)*.