Automatic Topic Labeling model with Paired-Attention based on Pre-trained Deep Neural Network

1st Dongbin He
College of Information and Electrical Engineering
China Agricultural University
Beijing, China
https://orcid.org/0000-0001-6197-5192

2nd Yanzhao Ren
College of Science
China Agricultural University
Beijing, China
xiaozhaochina@163.com

3rd Abdul Mateen Khattak
College of Information and Electrical Engineering
China Agricultural University
Beijing, China
mateen@aup.edu.pk

4th Xinliang Liu
College of Information and Electrical Engineering
China Agricultural University
Beijing, China
569489372@qq.com

5th Sha Tao
College of Information and Electrical Engineering
China Agricultural University
Beijing, China
taosha20070608@163.com

6th Wanlin Gao
College of Information and Electrical Engineering
China Agricultural UniversityBeijing, China
https://orcid.org/0000-0002-4845-4541

2021 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-3900-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/IJCNN52387.2021.9534093

***Abstract*—The automatic topic labeling model aims at generating a sound, interpretable, and meaningful topic label that is used to interpret an LDA-style discovered topic, intending to reduce the cognitive load of end-users while browsing or investigating the topics. In this study, we first introduced the pre-trained language model BERT to topic labeling tasks. It exploits the contextual embedding of the pre-trained language model to improve the quality of encoding sentences. To generate a topic label with higher Relevance, Coverage, and Discrimination, we propose a novel summarization neural framework. Specifically, it exploits the paired-attention to model the relationship between the candidate sentences first and then decides which sentences should be included in the final summarization topic label. Moreover, we expected that high-quality sentence encoding representation could improve our model's performance. So, for each discovered topic, we trained a specific layer to extract the important topic-related features from the sentence embeddings as well as filter the noise information. The experimental results showed that our model significantly outperforms the state-of-the-art and classic topic labeling models.**

Keywords—paired-attention, pre-trained Deep Neural Network, topic model, topic label, Latent Dirichlet Allocation (LDA)

I. INTRODUCTION

The topic model is a very important technique in natural language processing, such as information retrieval and text mining, and the discovered topics are usually represented by a list of topic terms with marginal probability [1]. Even if word distribution makes intuitive sense, it is still very difficult to understand what a topic really means and what is the difference between topics [2]. Therefore, the cognitive overhead in interpreting the topic discovered terms can be high, especially when users lack background knowledge in the field of topics [3].

To reduce the cognitive overhead of understanding these topics for users, the application of automatic topics labeling technique to generate meaningful labels is receiving more attention and has become a very challenging work. Some of the existing studies on topic labeling task exploit phrases, abstracts, or images to enhance the interpretation, e.g., generating topic label with phrases [2, 4-13], generating topic label with summaries [3, 14-16], and using images to interpret topics [8, 17-19]. However, the phrase-based labels are too short to adequately interpret the topics. Moreover, using images to illustrate topics is only appropriate under certain conditions. To fully interpret the topics, Basave et al. [14] first introduced a framework that applied summarization methods to generate topic labels (usually a paragraph consisted of the salient sentences). Their algorithms are independent of external sources (e.g., Wikipedia, WordNet) and only rely on the identification of dominant terms relevant to the latent topic [14].

To generate topic labels with high Relevance, Coverage and Discrimination, Wan and Wang [3]proposed a novel two-stage textual summary method based on submodular optimization. They observed that the use of summaries as labels had obvious advantages over the use of words and phrases. He et al. [16] proposed a novel graph-based ranking model that applies a specific strategy that suppresses or enhances the voting ratios of the vertices (sentences) in the ranking process to restrain the redundancy and boost diversity of the topic labels generated.

However, all the prevailing methods rely only on handcrafted features. There are two common pitfalls in the topic labeling task. First, exact word matching suggests that different words must have different meanings, which may damage the accuracy of the topic modeling. Secondly, it is hard to capture the real meanings of words or sentences without contextual information.

To address this problem, Bhatia et al. [12] proposed a simple method. They first used Wikipedia document titles as candidate labels, and then calculated the neural embeddings of documents and words to select the labels most relevant to the topics. To get

978-0-7381-3366-9/21/$31.00 ©2021 IEEE

the embedding of sentences and words, they trained a doc2vec model on the English Wikipedia corpus, while the word embedding was generated using word2vec during the doc2vec internal training process. The result showed that their approach was able to find a succinct phrase label for each topic, and outperformed other compared topic labeling systems.

Text summary achieves better performance through the contextual embedding rather than using the handcrafted features of texts only [20]. Recently, several methods have been proposed to build neural embeddings, such as word2vec [21] and doc2vec [22], and BERT [23].

To the best of our knowledge, we are the first to investigate and propose an extractive summarization topic labeling model based on a BERT layer. With its powerful architectures and pre-training on a huge corpus, BERT learned very complex contextual features that were used to improve the quality of encoding words or sentences that helped our model to generate better topic labels. Inspired by Liu [24], we transform the topic labeling process into a single-document summarization task. The detail of this process is provided in Section III. Method.

To further improve the quality of topic labeling, we proposed a novel **Topic Labeling** model with a **Paired-Attention** (**TLPA**), consisting of three layers: BERT layer, extracting layer, and summarization layer.

The BERT layer is located at the bottom, which is used to encode the input sentences. On top of the model, there is an extractive summarization layer to extract the appropriate sentence to composite the summarization topic label. The output of the BERT layer is the contextual embeddings of sentences, which can be directly fed to the top layer to learn how to generate topic labels. To improve TLPA's performance, we trained an additional middle layer that, from the BERT layer, extracts the important features and filters the noise information. In Section III. Method E. Summarization Layer, we detailed the extracting layer which had acquired the ability to refine the BERT embeddings of sentences after a pre-training task.

Extensive experiments have been conducted on AP and SIGMOD corpora [3, 4, 16]. The results show that TLPA can extract salient sentences and generate meaningful topic labels with minimum redundancy. Besides, it significantly outperforms the prevailing state-of-the-art and classic models. Our main contributions in this study are as follows:

1) We first introduced the pre-trained language model BERT to topic labeling tasks. It encoded each candidate sentence to the contextual embedding that helps TLPA effectively to overcome the limitation of simply relying on the handcraft features only.

2) We trained an extracting layer, which uses the transformer encoder to deal with the sentence BERT embeddings. It extracts out the important topic-related features and filters the noise information (irrelevant part). The high-quality representation obtained could effectively improve the performance of our model.

3) In the summarization layer, we constructed a positive and a negative inter-sentence encoder to simulate the attentive making decision of a human in opposite direction. There are two opposite attention score sentences simultaneously. The sentence wins more award points if it is most representative and inclusive, and suffers maximum penalty points if it is irrelevant and unimportant. The results showed that using the paired-attention significantly improved the model scoring accuracy compared with the single-attention.

The rest of the paper is arranged as follows: Section II presents the problem definition. In Section III, we introduce our topic labeling model containing three layers i.e., BERT, extracting, and summarization. Section IV describes the experimental settings and details, followed by the Result and Discussion in Section V, and finally, in Section VI, we present our conclusions and future work.

II. Problem Definition

A. Discovered Topic By LDA Model

In this study, we set k as the number of topics learned from the LDA model. It should be fixed a-priori [25] , and the Gibbs algorithm is the estimation method to fit the model. The V represents a vocabulary set of the corpus, w denotes a word in V, and topic T is a list of topic terms ranked by conditional probability $\{._T(w)\}_{w\in V}$. For each topic T, it has the $\sum_{w\in V} p_T(w) = 1$ [7]. Following Wan's study [3], we use the top 500 terms to represent the discovered topic because the sum of the conditional probabilities of the top 500 terms is close to 1.

B. Topic embedding

To obtain the more topic-related sentences, except the existing top-500 topic terms list of the given topic, we have provided another topic representation, called topic embedding, which transformed the top-500 topic terms into continuous feature vectors. The equations are described as follows:

$$N_{ts} = \sum_{T\in\emptyset}\sum_{w}\left|w \in top500(T)\right| \tag{1}$$

$$t_i = \log\left(\frac{|T|+1}{\sum_{T\in\emptyset}|w_i \in T|}\right) * P_T(w_i) \tag{2}$$

$$TopicEmb(T) = \left[t_0, t_1, \ldots \ldots t_i, \ldots \ldots, t_{N_{ts}-1}\right] \tag{3}$$

Where Ø is the set of all discovered topics, the dimension of the topic embedding is N_{ts} that is the number of the categories of all different top-500 topic terms, as shown in Eq. (1). *TopicEmb*(T) is the topic embedding of a given topic T, t_i is the *i-th* component of the *TopicEmb*(T) which is corresponded to the *i-th* term w_i of the topic T, as shown in Eq. (3). $P_T(w_i)$ is the conditional probability of w_i belonging to topic T, and $|T|$ represents the number of discovered topics, as shown in Eq. (2).

Especially, we used the topic embedding to train the extracting layer that predicted the KLD score between each sentence and the given topic (refer to Section III. Method C. BERT layer).

III. Method

A. Overview

Inspired by previous studies [3, 24, 26, 27], we introduced a two phases topic labeling model, TLPA:

1) First phase

we applied Wan and Wang [3] approach to score sentences of the current corpus for each discovered topic. They reported that most sentences in the corpus were irrelevant to the given

topic. Thus, the needs was to extract the top-500 sentences from AP/SIGMOD for a given topic, and then consider them as Candidate Sentences Set (*CSSet*). This way, the collection of all CSSet belonging to the different topics is called *CSSets*.

Bigi [28] argues that the Kullback-Leibler Divergence (KLD) method outperforms the conventional methods involving the tf-idf method. Wan and Wang [3] use KLD to measure the relevance between sentences and the discovered topics. The equation is defined as follows.

$$KLD(T,S) = \sum_{w \in S \cup T} P_T(w) * \log \frac{P_T(w)}{tf(w,S)/|S|} \quad (4)$$

Where $KLD(T,S)$ represents the KLD between topic T and sentence S, $tf(w,S)$ represents the frequency of word w in S, and $|S|$ denotes the word count of S. According to Wan and Wang [3], if a word w is not in sentence S, the $tf(w,S)/|S|$ would be replaced with 0.00001.

After the scoring process, we obtain the KLD between each sentence and each discovered topic in the current corpus. It can be used as the golden-standard data to train the extracting layer (refer to Section III. Method D. Extracting Layer).

2) Second phase

We used a novel summarization approach to label discovered topics if the candidate sentences are viewed as a single document. The extractive summarization of a single document proposed by Liu et al. [24, 29] is currently the most successful method based on the pre-trained model BERT. Inspired by their work, we proposed a BERT-based summarization model to label the discovered topics. This transformed a topic labeling task into a textual summarization task. Specifically, in the process of generating a topic label, we observe the issue of whether a sentence is included in the final summarization topic label as the optimization problem of binary classification. Our model consists of three parts, as shown in Fig. 1.

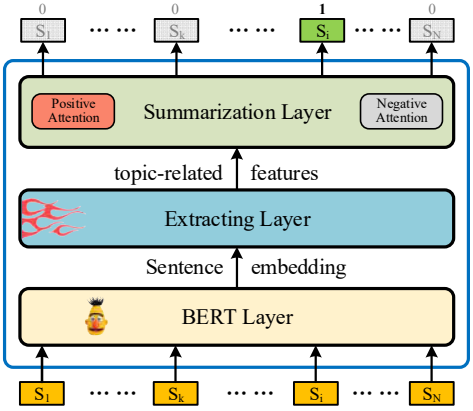

Fig. 1. The three layers of TLPA: the bottom layer is a pre-trained BERT that operates at the sentence level; the top layer is a classifier that runs over document level, which summarizes the discovered topics with paired-attention; the middle layer is a sentence features extractor that runs at the sentence level.

B. Training data

That lack of annotation data has always been a prominent problem in NLP tasks. We also face the same question in our training process. To acquire high-quality training data having the same distribution, we followed the random sampling method proposed by Ren et al. [27] to adaptively generate the training sample document D. The rules of generating D are described as follows.

1) In the first step, depending on a random number, it is determined that sampling sentences come from *CSSets* or the set of all sentences in the current corpus (*CorpusSets*). This is shown in the following equation.

$$\begin{cases} CSSets & rnd(0,1) > 1 - P_{sel} \\ CorpusSets & Otherwise \end{cases} \quad (5)$$

Where $rnd(0,1)$ function generates a random number from a uniform distribution within the range [0, 1]. P_{sel} is the given threshold. *CorpusSets* is a set that contains all sentences in the current corpus.

2) In the second step, the sentence will be selected if it has a lower KLD value. Besides, each sentence has a chance (at least 5%) of being chosen unconditionally, as shown in Eq. (6). After selecting 100 sentences, a sample document D is generated, before jumping to the next step.

$$\begin{cases} NotSelected & rnd(0,1) > 0.05 * Norm(1 / KLD(T,S)) + 0.05 \\ Selected & Otherwise \end{cases} \quad (6)$$

Where $KLD(T,S)$ represents the KLD value between the sentence S and the given topic T, $Norm(x)$ is a normalized function with the range [0, 1].

3) In the last step, according to the limited summary length, we exploit the greedy approach to find and tag the oracle sentences of each D, and then repeat the three steps until an adequate tagged training data set is available.

The training process quickens if the training sample document collections contain more sentences with higher conditional probability [27]. Besides, our experimental results also demonstrate the effectiveness of their approach to generate training sample document collections.

C. BERT layer

Wei et al. [30] suggested that BERT can learn very complex contextual features and can be used to improve the performance of the summarization task. Thus, we used the pre-trained language model BERT to encode the input sentence to contextual embedding.

Document D consisted of N sentences $[S_1, S_2, \ldots, S_N]$, as a batch of data, is fed to the BERT layer. The BERT encoded the D to a vector $X=[X_1, X_2, \ldots, X_N]$, which can be viewed as a batch of sentences embeddings, while the X_i represents *i-th* component of X corresponding to the *i-th* sentence S_i of D.

We inserted token [CLS] before each sentence and appended token [SEP] at the end. Mostly, in practice of sentence encoding that use BERT [23, 24, 31, 32], the token [CLS] is used to aggregate the features from a sentence. For a sentence S in the document D, the corresponding vector X_S is represented by the token [CLS] from the top BERT output embeddings.

D. Extracting Layer

For a summarization topic labeling model based on the pre-trained BERT (Fig. 1), the middle extracting layer is not necessary. The output vector of the BERT layer can be directly

fed to the summarization layer to generate topic labels. However, to generate a better Relevance, Coverage, and Discrimination topic label, we should provide the topic-related features of each candidate sentence to the summarization layer. That can help TLPA to select sentences accurately and control redundancy easily. So we propose a specific extracting layer to learn how to extract the useful features from the dense vector of each sentence based on the attention mechanism. Training a feature extractor in advance can accurately extract the important features related to a given topic from the embedding of sentences. This has a significant effect on improving the accuracy of binary classifiers in the summarization layer.

In the present study, there are two different versions of the extracting layer:

In the first version, we established corresponding multi-head attention (Transformer Encoder) for each discovered topic. The Q, K, and V inputs are the same sentence embedding in the training process, as shown in Fig. 2A.

In the second version, we set up one shared multi-head attention to extract features for each discovered topic. Zhou et al. [33] suggested that Multi-head Attention can be viewed as a feature extractor, which extracts the information from a set of key-value pairs (K, V) when given a query vector Q. Thus, we make the topic embedding (see Eq. 3) as Q, while K and V are still the same BERT embedding sentences, as shown in Fig. 2B.

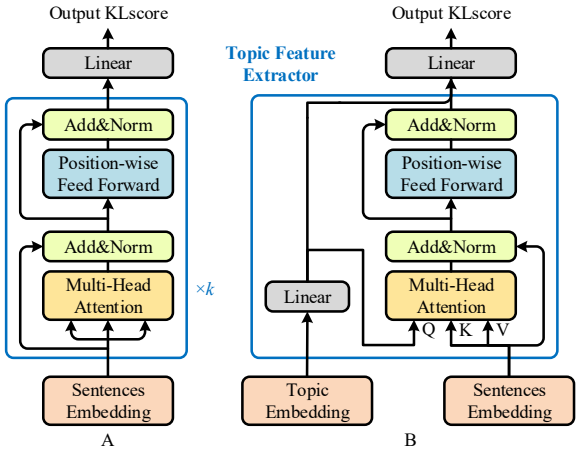

Fig. 2. Two implementation versions (A and B) of Sentence Feature Extracting Layer

According to Fig. 2A, the extracting layer has *k* multi-head attention, where *k* is the amount of discovered topics. It means that each discovered topic corresponds to a features extractor. In the topic labeling task, there is no positional relationship between the sentences in the training corpus. So, the positional embedding is not available in our extracting layer.

We trained the extracting layer to predict the KLD value between each sentence and each topic. It applies a simple linear layer as the decoder to transform the sentence features vector into a single value (the salience score of *S*), as shown in Eq. (7).

In the training process, the extractor keeps learning how to effectively extract topic-related features of the sentence while filtering out features related to other topics. In addition, use the golden-standard data to train the extracting layer that comes from the process of generating *CSSet* for each topic (see Section III. Method B. Training data).

$$KLDscore(S \mid T) = \frac{W_{KL} f(S \mid T) + b_{KL}}{\sqrt{d_S}} \tag{7}$$

$$f_A(S \mid T) = \mathrm{FFN}(Dropout(\mathrm{MHAtt}(emb(S), emb(S), emb(S) \mid T)) + emb(S)) \tag{8}$$

$$f_B(S \mid T) = \mathrm{FFN}(Dropout(\mathrm{MHAtt}(TopicEmb(T), emb(S), emb(S) \mid T)) + emb(S)) \tag{9}$$

$$\mathrm{FFN}(X_i) = Dropout(W_2(Dropout(gelu(W_1 LN(X_i) + b_1) + X_i) + b_2)) + X_i \tag{10}$$

$$\mathrm{MHAtt}(Q, K, V \mid T) = [head_1(Q, K, V \mid T), \ldots, head_7(Q, K, V \mid T)]W^O \tag{11}$$

$$head_j(Q, K, V \mid T), = \mathrm{Attention}(QW_j^Q, KW_j^K, VW_j^V \mid T) \tag{12}$$

Where d_S is the dimension of S_j and $1/\sqrt{d_S}$ is the scaling factor [33, 34], which is used to prevent dot products from growing too large in magnitude, $emb(S)$ represents BERT embedding of sentence S, $f_A(S|T)$ is an extracting function of the extracting-layer-A (see Fig. 2A); MHAtt($Q,K,V|T$) denotes a multi-head attention operation with 8 heads [24, 29] for a given topic T; FFN(X_i) represents a Position-wise Feed Forward layer, '[]' denotes vector concatenation, and Attention represents a single-head attention for a given topic T [34].

The loss of the extracting layer is the Mean Square Error (MSE) Loss of prediction $KLD_{score}(S|T)$ against gold KLD value (see Eq. 4). This way, it will be more accurate and targeted to model the relevance and redundancy of the candidate sentences. The outcome shows that the additional extracting layer can effectively improve the performance of the summarization layer.

E. Summarization Layer

Wood [35] reported that using pros and cons for evaluation can be used to assist decision-makers to choose the best option, and Borji [36] exploited it to evaluate a GAN. Thus, we used a paired-attention to evaluate each sentence for bonus and penalty points separately in our summarization layer. The two opposite attention simulate the human behavior of considering two aspects while making decisions. The positive and negative differences can be added up to combine the overall difference in each value range, and more accurate assessment results can be obtained [35].

In Fig. 3, the paired-attention has two document-level inter-sentence encoders [24], Positive Attention and Negative Attention. Both inter-sentence encoders model the relationship between the sentences in a single document D to find representative sentences and control redundancy.

Following the studies of Liu [24, 29], we added four linear layers and used a sigmoid function to predict the score vector $\hat{Y} = [y_1, y_2, \ldots, y_i, \ldots, y_N]$ of document D, where y_i decides whether or not the *i-th* sentence belongs to the summarization topic label. It is quite intuitive that Positive Attention prefers to give a higher score to the sentences that have a higher Relevance with the given topic and have a strong representation of others, whereas Negative Attention did the opposite operation.

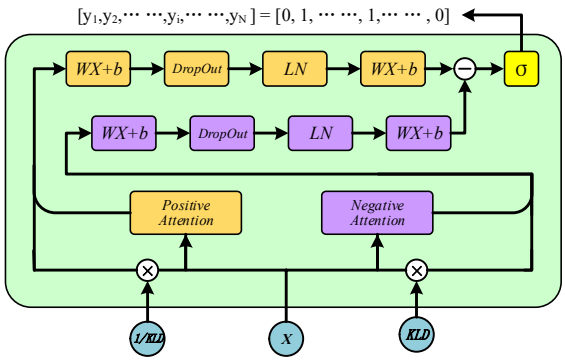

Fig. 3. Summarization Layer with Positive Attention and Negative Attention

Where *LN* is the layer normalization operation [37], σ is the Sigmoid function, X is the features vector that comes from the extracting layer or BERT Layer, and *KLD* is a vector of the Relevance (KLD value) of X with the given topic T.

Liu [24] figured out that the Transformer and Recurrent Neural Networks (e.g., LSTM) had their own advantages. Therefore, we implemented two different versions of the inter-sentence encoder, as shown in Fig. 4.

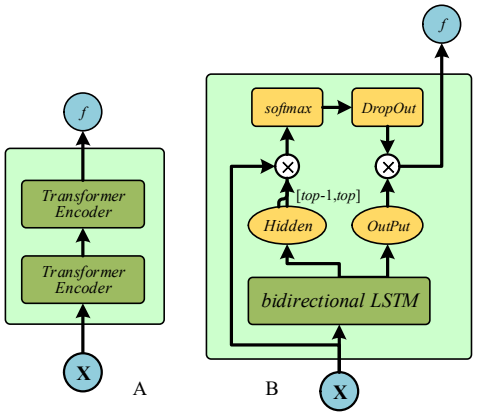

Fig. 4. A. Inter-Sentence Encoder consisted of double Transformer encoder stacked and B. Inter-Sentence Encoder constructed of a bidirectional and two layers LSTM.

Using the summarization layer, we transform a topic labeling (sentence selecting) task into an optimization problem of binary classification. For each sentence S_i in document D, we calculate the final predicted score $\hat{Y}_i$, and the equation is shown as follows,

$$\hat{Y}_i = \sigma(Weight(PosAtt(X_i)) - Weight(NegAtt(X_i))) \quad (13)$$

$$Weight(Att(X_i)) = W_o(LN(DropOut(W_f Att(X_i) + b_f))) + b_o \quad (14)$$

Where σ is the sigmoid function, X_i is a features vector of the *i-th* sentence embedding. There is no need to add positional embeddings [24] to X because there is no contextual relationship between the candidate sentences in D. $Weight(Att(X_i))$ calculates the weight of vector X_i by Positive or Negative Attention, shown as Eq. (14); *LN* represents Layer Normalization operation, $Att(X)$ denotes the features vector extracted by the Inter-Sentence Encoder (Positive or Negative Attention).

The Positive or Negative Attention can be implemented by Transformer Encoder or bidirectional LSTM. Thus, we derive the following equations, shown as Eq. (15-19).

$$PosAtt_{Tran} = [TranEncoder(X), (1/KLD) * X] \quad (15)$$

$$NegAtt_{Tran} = [TranEncoder(X), KLD * X] \quad (16)$$

$$PosAtt_{LSTM} = [Feature(X), (1/KLD) * X] \quad (17)$$

$$NegAtt_{LSTM} = [Feature(X), KLD * X] \quad (18)$$

$$Feature(X) = matmul(DropOut(softmax(matmul(X, [h^f, h^b])), output) \quad (19)$$

Where $PosAtt_{Tran}$ and $PosAtt_{LSTM}$ are the Positive Attention implemented by Transformer Encoder and bidirectional LSTM respectively, $NegAtt_{Tran}$ and $NegAtt_{LSTM}$ are Negative Attention implemented by Transformer Encoder and bidirectional LSTM respectively, $TranEncoder(X)$ represents the output of the Transformer Encoder. For the bidirectional LSTM encoder, the $\boldsymbol{h^f}$ and $\boldsymbol{h^b}$ are corresponding to the forward and backward hidden states, and ***output*** is the last layer of its output tensor.

To optimize the summarization layer [24, 29], the loss we used is the Binary Classification Entropy of $\hat{Y}_i$ against gold label Y_i and the equation is described as follows,

$$Loss = -\frac{1}{N}\sum_{i=1}^{N}(Y_i \log p(\hat{Y}_i = 1 \mid T) + (1 - Y_i)\log(1 - p(\hat{Y}_i = 0 \mid T)) \quad (20)$$

Where N is the number of sentences of D, $Y_i \in \{0,1\}$ is the ground-truth tag for sentence S_i of D, $Y_i = 1$ indicates selecting the sentence S_i into the final summarization topic label.

IV. Experiment

A. Experiment Setup

1) Data Sets

Following previous studies [3, 4], we used the two public document collections, i.e., SIGMOD and APNews. After a series of pretreatments, some unnecessary elements such as punctuations, digital numbers and stop words have been removed. Besides, words that are not nouns, verbs, adjectives, adverbs, or pronouns have been filtered out, such that the remaining words are all in the stemmed form. In total, there are 3016 and 2246 documents, 22105 and 46043 sentences, 8252 and 25902 vocabularies in the SIGMOD and APNews respectively.

According to the approach described in Section III. Method B. Training data, we obtained 24,000 training documents and 6,000 test documents for AP and SIGMOD collections respectively. Additionally, each training or test document D had 100 sentences, and the oracle sentences were tagged. To fairly compare with supervised models, we only trained and validated TLPA with the first 20 discovered topics, and used the rest to test all the contrast topic labeling models on *CSSets*.

2) Model implementation

In this paper, we exploited the Gensim to discover topics in LDA manner, and used Pytorch and the 'bert-base-uncased' version of BERT to build our model. The parameters are borrowed from prior studies or empirically set, e.g., to discover the topics, it is necessary to fix the topic number k=25 [3]. According to the recommended length of the summary in DUC

conferences [27], the length of the summary topic label was limited to 100 words.

Based on the consideration of efficiency and low-resource requirement, we did not use fine-tuned BERT, and all models were trained for 60,000 steps on a GPU(RTX2060S). In our training process, the batch size was 100, and the learning rate schedule of optimizer Adam was following [24] with warming-up on first 4,000 steps. Besides, we set P_{sel}=0.5 in Eq. (5).

B. Labeling Models

To demonstrate the effectiveness of our approach, we compared TLPA with the following classical and state-of-the-art models: Lexrank [38], Textrank [32], Submodular [3], and TLRank [16]. Besides, we can use BertSum [24] to generate a summarization topic label if viewing the *CSSet* as a single document and not considering the positional relationship between sentences of the *CSSet*, thus we proposed a revised version of BertSum with None of the Position consideration (BertSum-NP). To investigate the effectiveness of the extracting layer, we built a comparison model without extracting layer, called TLPA has None of Extracting Layer (TLPA-NEL).

V. Results And Discussion

According to Wan and Wang [3] and Mei et al. [4], the topic summary generated for each topic should satisfy the three criteria of higher Relevance, Coverage, and Discrimination.

A. Result

1) Relevance

The higher relevance and lesser the redundancy, the better the summary topic labels are. Peyrard [39] points out that KLD integrates relevance and redundancy well, which is a perfect measure tool to evaluate the generated summary text.

For each topic labeling model, we first computed the KLD value between each generated topic label and the corresponding discovered topic, and then averaged all the KLD values in the case of AP and SIGMOD. The results (Table I) revealed that our model (TLPA) had the lowest value in each case and outperformed other competitors.

TABLE I. Comparison of the average of KL Divergence between summarization topic label and corresponding discovered topic

model	Relevance	
	AP	***SIGMOD***
LexRank	4.08042	3.18463
TextRank	3.92700	3.25402
Submodular	3.30045	2.73337
TLRANK-C	2.85353	2.04545
TLRANK-G	2.86216	2.03687
BERTSum-NP	3.20660	2.39269
TLPA-NEL	3.04770	2.05666
TLPA	**2.63031**	**1.82040**

Based on the pre-trained language model BERT, BERTSum-NP uses neural contextual embedding to represent candidate sentences. It surpasses the most unsupervised approaches that rely on handcraft features, e.g., LexRank, TextRank, and Submodular.

Compared with BERTSum-NP, the improvement of TLPAs (TLPA-NEL and TLPA) was obvious. Nevertheless, TLPA-NEL and BERTSum-NP are still behind TLRank because they are not like TLRank or TLPA to exploit the topic-related features of the candidate sentences in the labeling task. TLPA outperforms others because it uses the extractive layer to filter the irrelevant noise information, and extract more topic-related features of the sentences.

2) Coverage

According to Wan and Wang [3], Coverage is defined as the ratio of words that appeared in the top 20 topic terms. It chooses the top-20 terms instead of 500, because the top-20 terms are more significant and representative than the rest. In general, higher Coverage denotes more top topic terms covered by the topic label and is more comprehensive.

We average all the Coverage values in the case of AP and SIGMOD. The results presented in Table II show that TLPA has the best overall performance of covering the top-20 topic terms. It has a maximum coverage ratio of 0.26 for AP and ranks second with a coverage ratio of 0.23 for SIGMOD.

TABLE II. A comparison of the mean ratio of the words covered out of top 20 topic terms for each topic

model	Coverage	
	AP	***SIGMOD***
LexRank	0.15	0.18
TextRank	0.17	0.13999
Submodular	0.14	0.14
TLRANK-C	0.23	0.25
TLRANK-G	0.23	**0.25**
BERTSum-NP	0.19	0.19
TLPA-NEL	0.19	0.21
TLPA	**0.26**	**0.23**

3) Discrimination

The same sentence may have different probability distributions for each topic. Therefore, we should try our best to avoid the same or similar sentences appearing in different topic labels. Thus, the smaller the similarity between the topics labels, the better the quality of the topics labels is. To obtain Discrimination, we compute the cosine similarity between two different topic labels and then average all the similarity values. As obvious from Table III, Submodular that has a minimum Discrimination value, is the best in all cases, and TLPA ranks second.

TABLE III. A comparison of the means of all cosine similarity values among different topics labels

model	Discrimination	
	AP	***SIGMOD***
LexRank	0.03007	0.05963
TextRank	0.03744	0.07735
Submodular	**0.00729**	**0.01661**
TLRANK-C	0.02711	0.04743
TLRANK-G	0.0277	0.04988
BERTSum-NP	0.03416	0.05411
TLPA-NEL	0.02119	0.05599
TLPA	**0.01849**	**0.04599**

According to Tables I, II, and III, from the perspective of Relevance, Coverage, and Discrimination, TLPA achieves the best comprehensive performance compared with other models.

B. Discussion

1) Effectiveness of Paired-Attention and Extracting Layer

Inspired by Liu [24], we introduced TLPA, a novel summarization topic labeling model. To further improve the performance of this model, we added an extracting layer to capture more topic-related sentence features and filter out irrelevant information. Besides, paired-attention was built in the summarization layer to simulate human scoring sentences from the perspective of positive and negative, which effectively improves the accuracy and refrains redundancy in the generating process.

There are two implementation versions (A and B) of the extracting layer defined in Section III Method (see Fig. 2). The sentence topic-related features extracted by the two versions extracting layers denoted as Topic-related Features 1 (TF1) and Topic-related Features 2 (TF2) respectively. Besides, we have two versions (Transformer and LSTM) of the summarization layer. Therefore, it obtained four training models: TLPA-TF1-Transformer, TLPA-TF1-LSTM, TLPA-TF2-Transformer, and TLPA-TF2-LSTM.

To further understand the improvement effects provided by paired-attention and extracting layer, we compare the above four training models with the two implementation versions (Transformer and LSTM) of BertSum-NP and TLPA-NEL, as shown in Table IV below:

TABLE IV. RELEVANCE OF TOPIC LABELS

model	AP			SIGMOD		
	Min	*Max*	***Mean***	*Min*	*Max*	***Mean***
BERTSum-NP-Transformer	1.98900	4.2642,	3.20660	1.77186	3.07262	2.39269
BERTSum-NP-LSTM	3.08190	5.17400	3.67200	2.01950	3.94035	2.68760
TLPA-NEL-Transformer	2.27330	3.84050	3.04770	1.94753	2.25155	2.05666
TLPA-NEL-LSTM	2.39510	5.12730	3.49220	1.79078	2.79571	2.23868
TLPA-TF1-Transformer	2.21433	3.04437	2.80142	1.48039	2.41418	1.93894
TLPA-TF1-LSTM	2.04369	3.95774	2.99373	1.55924	2.41983	1.88100
TLPA-TF2-Transformer	1.97160	3.08899	**2.63031**	1.50106	2.25590	**1.82040**
TLPA-TF2-LSTM	2.15079	3.10607	2.69212	1.60251	2.35287	1.89157

According to Table IV, in most cases, there is a small difference in Relevance between the Transformer version and the LSTM version, while the Transformer has better performance. That is consistent with the trend we saw in Fig. V, VI, and VII. Besides, TLPA-TF2-Transformer achieved the best performance in the case of both AP and SIGMOD. So, we suggest that paired-attention and extracting layer play a key role to improve the Relevance of TLPA.

According to the results in Table V, the Transformer version still has an absolute advantage over the LSTM version in terms of Coverage, but the difference is not very large. Not surprisingly, TLPA-TF2-Transformer still performed best both in the case of AP and SIGMOD. Besides, as obvious from the changes of Coverage, paired-attention and extracting layer have a clear effect in improving Coverage.

An identical conclusion can be drawn from Table VI; the Transformer version has an absolute advantage over the LSTM version in terms of the Discrimination between topic labels. Whether in the case of AP or SIGMOD, the reason for TF1-Transformer version being more successful maybe that it does not have shared parameters. In TF1-Transformer, each topic has its Transformer encoder unit, so it is better to grasp the topic difference. Although TLPA-TF2-Transformer only ranks third after TLPA-TF1-Transformer and TLPA-TF1-LSTM, the actual difference is very small (an absolute difference not more than 0.0015). Finally, it is easy to find that the paired-attention and extraction layer plays a very important role in optimizing Discrimination and has an overall positive effect.

TABLE V. COVERAGE OF TOPIC LABELS

model	AP			SIGMOD		
	Min	*Max*	***Mean***	*Min*	*Max*	***Mean***
BERTSum-NP-Transformer	0.05	0.40	0.19	0.15	0.20	0.19
BERTSum-NP-LSTM	0.05	0.35	0.14	0.05	0.25	0.17
TLPA-NEL-Transformer	0.05	0.40	0.19	0.10	0.25	0.21
TLPA-NEL-LSTM	0.05	0.30	0.15	0.10	0.25	0.20
TLPA-TF1-Transformer	0.05	0.45	0.23	0.15	0.25	0.21
TLPA-TF1-LSTM	0.05	0.45	0.19	0.15	0.30	0.21
TLPA-TF2-Transformer	0.05	0.45	**0.26**	0.20	0.25	**0.23**
TLPA-TF2-LSTM	0.10	0.40	0.23	0.20	0.25	0.22

TABLE VI. DISCRIMINATION OF TOPIC LABELS

model	AP			SIGMOD		
	Min	*Max*	***Mean***	*Min*	*Max*	***Mean***
BERTSum-NP-Transformer	0.01053	0.05682	0.03416	0.03448	0.10843	0.05411
BERTSum-NP-LSTM	0.01667	0.08738	0.03763	0.03226	0.16495	0.07036
TLPA-NEL-Transformer	0.01031	0.03125	0.02119	0.03125	0.09574	0.05599
TLPA-NEL-LSTM	0.01020	0.07527	0.03308	0.03261	0.12658	0.06490
TLPA-TF1-Transformer	0	0.04494	0.01693	0.01235	0.08537	**0.04449**
TLPA-TF1-LSTM	0	0.04854	0.01820	0.01010	0.08108	0.04517
TLPA-TF2-Transformer	0.00943	0.05208	**0.01849**	0.02273	0.10976	0.04599
TLPA-TF2-LSTM	0.10000	0.40000	0.23000	0.20000	0.25000	0.22000

2) Sensitivity of Batch-Size

According to the studies of Wan and Wang [3], the number of candidate sentences used to generate topic labels is equal to the number of sentences in *CSSet* (i.e., 500). It is also the batch-size in our prediction experiment. However, based on the consideration of efficiency and low-resource requirement, we found that batch-size less than 100 is acceptable during the training process.

In the predicting process, to understand the sensitivity of all the above labeling models to the input batch-size, and to inspect

the stability of the above labeling models under the different batch-sizes, we build a set of different batch-sizes, LenSet= {100, 150, 200, 250, 300, 350, 400, 450, 500}, for observing the changes of the Relevance, Coverage, and Discrimination.

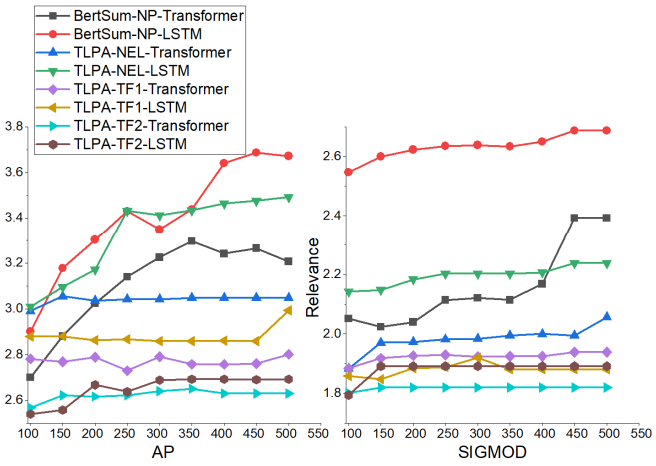

Fig. 5. The averaged Relevance of the different models under the different batch-sizes 100, 150, 200, 250, 300, 350, 400, 450 and 500

Fig. 5 reveals that for different models, the implementation versions of Transformer have better Relevance than LSTM, and the curves of trends are more stable. For example, Bertsum-NP-Transformer is better than Bertsum-NP-LSTM. Compared with the BertSum-NP model (having high sensitivity to batch-size), the novel paired-attention not only effectively reduces the KLD value but also makes the model more stable. Especially, due to an extracting layer, TLPA-TF2-Transformer achieved the best performance in Relevance, and its stability was also prominent. Finally, we can conclude that a model with both paired-attention and extracting layer is more stable (less sensitivity to batch-size of input).

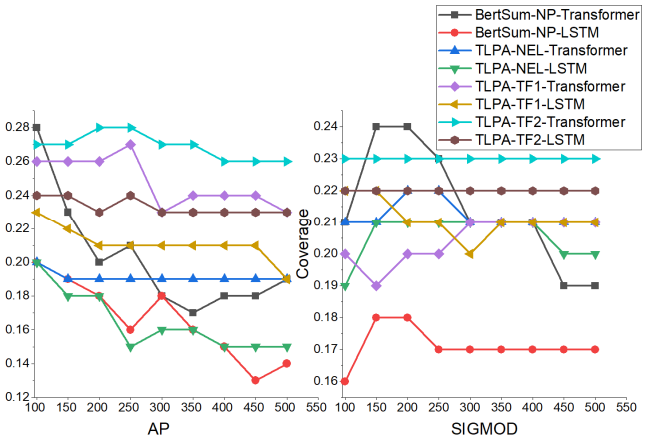

Fig. 6. The averaged Coverage of the different models under the different batch-sizes 100, 150, 200, 250, 300, 350, 400, 450 and 500

According to Fig. 6, for Coverage, the Transformer versions of each model are better than LSTM and the curves of trends are relatively stable. For BertSum-NP, both implementation versions in AP show a sharp declining trend. In SIGMOD, except for TLPA-TF2-Transformer and TLPA-TF2-LSTM (both having relatively higher Coverage and stable trend), all the models have lower Coverage and fluctuate in different degrees. Especially, BertSum-NP-Transformer (fluctuating sharply) is very sensitive to the batch-size.

Compared with the BertSum-NP, the paired-attention and extracting layer have not only effectively enhance the Coverage of the topic label, but also reduced the sensitivity of the labeling model to the input batch-size. In the case of AP and SIGMOD, the TLPA-TF2-Transformer model not only provides the highest Coverage under most batch-sizes but also has the best stability.

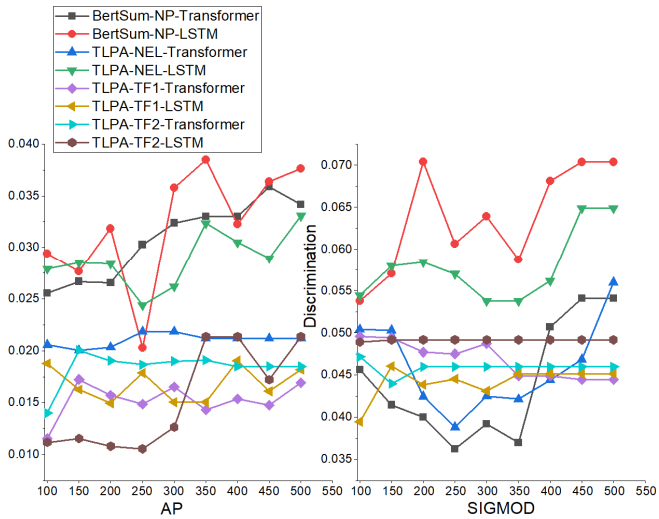

Fig. 7. The averaged Discrimination of the different models under the different batch-sizes 100, 150, 200, 250, 300, 350, 400, 450 and 500

As per Fig. 7, we found that all the absolute values of the averaged similarity are distributed within the range of 0.01 to 0.071. It means that all the topic labels generated by these contrasting models have very low similarity values and very good discriminability. Especially, TLPA-TF2-Transformer is the only model that is insensitive to batch-size variation, which can maintain a relatively stable curve trend both in the case of AP and SIGMOD, and its value is within a relatively reasonable range in the lower half of the overall value distribution.

Compared with the model of LSTM implemented version, Transformer has better performance in Relevance, Coverage, and Discrimination (see Fig. 5, 6, and 7). It means that Transformer is less sensitive to the input batch-size changes and has better stability.

VI. Conclusion and Future Work

In this paper, we proposed an automatic topic labeling model based on a pre-trained deep neural network. To the best of our knowledge, it is the first time to exploit the BERT to get a high-quality contextual feature to summarize the topics discovered. To obtain the really important features to identify the salient sentence, we added a middle layer, from the BERT embedding of the sentences, to extract the topic-related features and filter the noise information. Finally, we constructed a paired-attention to simulate the attentive making decision of a human in opposite direction, aiming at selecting the appropriate sentences to compose a meaningful and concise topic label. Besides, our extractive summarization topic label does not have the English grammar problem that it is hard to avoid in the topic label generating of abstractive manner. The results on the two datasets

showed that our model outperforms the prevailing state-of-the-art and previous classic models.

In future research, to improve the performance of the topic labeling model, we will try to increase the searching space of the *CSSets* to find that sentences having diverse information but being less relevant to the given topic. Further, we shall focus on how to effectively control redundancy and obtain the best combination of selected sentences that generate a summarization topic label with higher Relevance, Coverage, and Discrimination.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (Grant No. 31801669)

REFERENCES

[1] D. M. Blei, A. Y. Ng and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research,* vol. 3, pp. 993-1022, 2003-01-01 2003.

[2] D. Magatti, S. Calegari, D. Ciucci, and F. Stella, "Automatic labeling of topics," in *2009 Ninth International Conference on Intelligent Systems Design and Applications*, 2009, pp. 1227-1232.

[3] X. Wan and T. Wang, "Automatic labeling of topic models using text summaries," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 2297-2305.

[4] Q. Mei, X. Shen and C. Zhai, "Automatic labeling of multinomial topic models," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007, pp. 490-499.

[5] J. H. Lau, K. Grieser, D. Newman, and T. Baldwin, "Automatic labelling of topic models,", 2011, pp. 1536-1545.

[6] I. Hulpus, C. Hayes, M. Karnstedt, and D. Greene, "Unsupervised graph-based topic labelling using dbpedia," in *Proceedings of the sixth ACM international conference on Web search and data mining*, 2013, pp. 465-474.

[7] N. Aletras and M. Stevenson, "Labelling topics using unsupervised graph-based methods," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2014, pp. 631-636.

[8] N. Aletras, T. Baldwin, J. H. Lau, and M. Stevenson, "Representing topics labels for exploring digital libraries," in *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries*, 2014, pp. 239-248.

[9] W. Kou, F. Li and T. Baldwin, "Automatic labelling of topic models using word vectors and letter trigram vectors," in *AIRS*, 2015, pp. 253-264.

[10] M. Allahyari and K. Kochut, "Automatic topic labeling using ontology-based topic models," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, 2015, pp. 259-264.

[11] Z. Li, J. Li, Y. Liao, S. Wen, and J. Tang, "Labeling clusters from both linguistic and statistical perspectives: A hybrid approach," *Knowledge-Based Systems,* vol. 76, pp. 219-227, 2015-01-01 2015.

[12] S. Bhatia, J. H. Lau and T. Baldwin, "Automatic labelling of topics with neural embeddings," *arXiv preprint arXiv:1612.05340,* 2016-01-01 2016.

[13] A. Alokaili, N. Aletras and M. Stevenson, "Re-ranking words to improve interpretability of automatically generated topics," *arXiv preprint arXiv:1903.12542,* 2019-01-01 2019.

[14] A. E. C. Basave, Y. He and R. Xu, "Automatic labelling of topic models learned from twitter by summarisation," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2014, pp. 618-624.

[15] M. H. Barawi, C. Lin and A. Siddharthan, "Automatically labelling sentiment-bearing topics with descriptive sentence labels," in *International Conference on Applications of Natural Language to Information Systems*, 2017, pp. 299-312.

[16] D. He, M. Wang, A. M. Khattak, L. Zhang, and W. Gao, "Automatic Labeling of Topic Models Using Graph-Based Ranking," *IEEE Access,* vol. 7, pp. 131593-131608, 2019.

[17] N. Aletras and A. Mittal, "Labeling topics with images using a neural network," in *European Conference on Information Retrieval*, 2017, pp. 500-505.

[18] N. Aletras and M. Stevenson, "Representing topics using images," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 158-167.

[19] I. Sorodoc, J. H. Lau, N. Aletras, and T. Baldwin, "Multimodal topic labelling," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 2017, pp. 701-706.

[20] W. Zhao, M. Peyrard, F. Liu, Y. Gao, C. M. Meyer, and S. Eger, "Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance," *arXiv preprint arXiv:1909.02622,* 2019.

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781,* 2013.

[22] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*, 2014, pp. 1188-1196.

[23] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805,* 2018.

[24] Y. Liu, "Fine-tune BERT for Extractive Summarization," *arXiv preprint arXiv:1903.10318,* 2019.

[25] K. Hornik and B. Grün, "topicmodels: An R package for fitting topic models," *Journal of Statistical Software,* vol. 40, pp. 1-30, 2011-01-01 2011.

[26] R. Nallapati, F. Zhai and B. Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[27] P. Ren, F. Wei, C. Zhumin, M. A. Jun, and M. Zhou, "A redundancy-aware sentence regression framework for extractive summarization," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 33-43.

[28] B. Bigi, "Using Kullback-Leibler distance for text categorization," in *European Conference on Information Retrieval*, 2003, pp. 305-319.

[29] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," *arXiv preprint arXiv:1908.08345,* 2019.

[30] R. Wei, H. Huang and Y. Gao, "Sharing Pre-trained BERT Decoder for a Hybrid Summarization," in *China National Conference on Chinese Computational Linguistics*, 2019, pp. 169-180.

[31] R. Nogueira and K. Cho, "Passage Re-ranking with BERT," *arXiv preprint arXiv:1901.04085,* 2019.

[32] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004.

[33] Q. Zhou, N. Yang, F. Wei, S. Huang, M. Zhou, and T. Zhao, "A Joint Sentence Scoring and Selection Framework for Neural Extractive Document Summarization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 28, pp. 671-681, 2020.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, A. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998-6008.

[35] M. Wood, "The pros and cons of using pros and cons for multi-criteria evaluation and decision making," *Available at SSRN 1545189,* 2009.

[36] A. Borji, "Pros and cons of gan evaluation measures," *Computer Vision and Image Understanding,* vol. 179, pp. 41-65, 2019.

[37] J. L. Ba, J. R. Kiros and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450,* 2016.

[38] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of artificial intelligence research,* vol. 22, pp. 457-479, 2004-01-01 2004.

[39] M. Peyrard, "A simple theoretical model of importance for summarization," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 1059-1073.