electronics

MDPI

Article

Incorporating External Knowledge into Unsupervised Graph Model for Document Summarization

Tiancheng Tang [1], Tianyi Yuan [1], Xinhuai Tang [1,*] and Delai Chen [2]

[1] School of Software, Shanghai Jiao Tong University, Shanghai 200240, China; tangtian_cheng@sjtu.edu.cn (T.T.); gran_turismo@sjtu.edu.cn (T.Y.)

[2] Shanghai Key Lab of Advanced Manufacturing Environment, China Telecom Shanghai Branch, Shanghai 200030, China; dl.chen@163.com

* Correspondence: tang-xh@cs.sjtu.edu.cn

Received: 20 August 2020; Accepted: 13 September 2020; Published: 17 September 2020

Abstract: Supervised neural network models have achieved outstanding performance in the document summarization task in recent years. However, it is hard to get enough labeled training data with a high quality for these models to generate different types of summaries in reality. In this work, we mainly focus on improving the performance of the popular unsupervised Textrank algorithm that requires no labeled training data for extractive summarization. We first modify the original edge weight of Textrank to take the relative position of sentences into account, and then combine the output of the improved Textrank with K-means clustering to improve the diversity of generated summaries. To further improve the performance of our model, we innovatively incorporate external knowledge from open-source knowledge graphs into our model by entity linking. We use the knowledge graph sentence embedding and the tf-idf embedding as the input of our improved Textrank, and get the final score for each sentence by linear combination. Evaluations on the New York Times data set show the effectiveness of our knowledge-enhanced approach. The proposed model outperforms other popular unsupervised models significantly.

Keywords: document summarization; knowledge graph; Textrank; unsupervised model

1. Introduction

Document summarization is considered as a challenging task in natural language processing. This task aims at generating some short sentences to summarize the main points of a whole document. Document summarization methods are divided into two categories. The first one is extractive summarization which selects sentences from source documents as a summary. Another important type is abstractive summarization, which is more difficult than extractive summarization. New words and phrases are generated by abstractive summarization algorithms to express the main idea of an original document.

Many recent studies focus on deep learning mechanisms in order to improve the performance of models on the extractive or abstractive summarization task. Many sequence-to-sequence models [1–3] and attention-based models [4] are proposed for this task. BERT [5] is also used in existing deep learning models to achieve a better performance [6,7]. These studies have achieved promising results in document summarization.

However, most of the above deep learning models are supervised models, and a large amount of labeled training data is required for the training process. In reality, it is difficult to get so much high-quality data set with labeled documents and summaries in all domains. Compared with other natural language processing tasks such as text classification, the available labeled data set for document summarization is limited, and most of the labeled documents are about news. Therefore, it is necessary

to research on improving the performance of unsupervised summarization algorithms that require no labeled training set.

Previous studies on unsupervised models mainly focus on extractive summarization. Most of the popular algorithms are based on graph algorithms or text cluster algorithms. Among them, graph algorithms such as Textrank [8] and Centroid [9] are the most popular. In these studies, each sentence of a document is embedded into a vector by using tf-idf [10], skip-thought vectors [11] or BERT [5]. These sentences are represented as nodes of a graph, and the similarity of these sentences is calculated as the weight of edges. Graph-sorting algorithms such as Pagerank [12] are used to calculate the score of these nodes. Sentences that have the highest scores are selected as the abstract of a document.

Although these graph models have achieved good results, there are still some limitations. One problem of the Textrank-based algorithm is that the relative position between sentences is not considered in Textrank because documents are turned into undirected graphs. Another problem is that sentences of the highest scores are very likely to be similar, which affects the diversity of the generated summaries. Furthermore, due to lack of labeled training data, it is difficult for these unsupervised models to achieve as good performance as supervised ones.

In this paper, a knowledge-enhanced unsupervised model based on Textrank and K-means is therefore proposed to overcome all the above problems. In this model, tf-idf algorithm is used for sentence embedding and the similarity between sentences is obtained by cosine similarity. Our first approach to address these problems is to improve the performance of Textrank by multiplying the weight separately according to the relative position between sentences. Our second approach is to improve the diversity of generated abstracts. The output of K-means cluster algorithm and Textrank is combined for the selection of top k sentences. The sentences selected should be in a different cluster so that the diversity of our abstract is improved.

To further improve the performance of our model, we incorporate external knowledge from open-source knowledge graphs to our model. We use external knowledge to generate better summaries for the following reasons. The first reason is that we can acquire external domain knowledge of high quality from knowledge graphs. Thanks to the development of open-source knowledge graph and entity linking systems, we can map words to entities on knowledge graphs such as Wikipedia (https://www.wikipedia.org/) with high accuracy. The second reason is that tf-idf is a count-based algorithm which does not take synonyms into account. If we use entity linking systems, similar words may be linked to the same entity, and a polysemous word may be linked to different ones so that we get a better sentence embedding. The third reason is that the domain knowledge from knowledge graphs has been proven useful in supervised summarization methods [13,14]. According to their studies, it is more meaningful for a summarization model to generate an entity in a special domain than other words. As a result that there is little research on using external knowledge in unsupervised summarization models, we decide to investigate on incorporating external knowledge into unsupervised models in this paper. In our approach, we use an open-source entity linking system to link words to entities in Wikipedia. All the entities in our documents are discovered, and each sentence is transferred to an one-hot vector based on the frequency of entities. We use the same algorithm based on our improved Textrank to get another score for each sentence. These scores are combined by linear combination and the results of K-means is also considered in our model to ensure that the sentences for summarization are not similar with each other.

Our main contributions are as follows:

(1) We improve the performance of traditional unsupervised learning models by modifying the Textrank algorithm and combining the output of Textrank and K-means. Our model takes the relative position of sentences into account and improves the diversity of the generated abstracts.
(2) We innovatively incorporate external knowledge from knowledge graphs to our unsupervised model to improve the performance on summarization. A different embedding method based on entity linking is proposed in this paper. Our model runs the improved Textrank algorithm twice to incorporate the external knowledge from Wikipedia into our model.

(3) Experiments are conducted on the New York Times data set. The experimental results show that our knowledge-enhanced model outperforms other mainstream unsupervised models.

2. Related Work

Document summarization has been widely studied for many years. Extractive summarization methods choose a subset of sentences from the original text to summarize a document. These sentences are scored and rearranged by many machine learning or deep learning algorithms. Before we use these ranking algorithms, sentences should be embedded into vectors to determine their similarity. A commonly used sentence embedding method is td-idf [10], which computes both the frequency and the inverse document frequency of words. Recently proposed skip-thought vectors [11] and BERT [5] are also useful sentence embedding methods, and they are widely used in deep learning summarization models. Many popular unsupervised models for document summarization are based on graph algorithms such as Textrank [8] and Centroid [9]. In the Textrank algorithm, the score of a sentence is calculated by Pagerank, which is a popular graph-based ranking algorithm in many areas. Lexrank [15] uses a stochastic graph-based method to compute the relationship of sentences, and this method is essentially identical to TextRank. To further improve the performance of these unsupervised graph algorithms, the original Textrank algorithm is often modified for the target task. For example, Mallick et al. [16] modify the inverse sentence frequency-cosine similarity by giving different weights to different words. Although these graph models perform well in document summarization tasks, the relationship between sentences are not considered in these models, and these models are likely to select similar sentences as summaries. In addition to the graph-based algorithms, unsupervised text cluster methods such as K-means [17] and K-medoids [18] are also used in document summarization tasks, and the sentences selected by these methods are independent of each other. In this paper, both the popular Textrank graph model and the K-means cluster algorithm are used in our approach. We take the relative position of sentences into account in order to improve the performance of Textrank, and select sentences that are independent of each other as summaries with the help of K-means cluster algorithms.

Deep learning models are also widely used in document summarization tasks. Many previous studies focus on neural sequence-to-sequence models for extractive and abstractive summarization [1,2]. See et al. [3] propose a pointer-generator network to overcome the shortcomings of traditional Seq2Seq models. Some studies focus on the use of reinforcement learning [19–21] in document summarization tasks and achieve good performance. Attention mechanism, Transformer [22] and BERT [7,23] are also popular methods in text summarization. Paulus et al. [4] propose a network with a novel intra-attention for the abstractive summarization of long documents. Bouscarrat et al. [6] propose a light model based on Transformer for extractive summarization. However, most of these deep learning mechanisms are supervised models and they need a large number of training data. In reality, we may not have enough labeled data to train these models.

Due to the lack of relevant domain knowledge, the performance of previous models is not satisfactory. To solve this problem, some recent studies focus on incorporating external knowledge into deep learning models to make these summarization models knowledgeable, and the external knowledge may come from templates, topics or knowledge graphs. Wang et al. [24] propose an abstractive model based on templates and a bi-directional selective mechanism. Different types of templates are used as external knowledge to generate better summaries. Zhou et al. [25] experiment on encoding distinctive external features such as Named Entity Recognition (NER) in the question generation task. The topics of documents are also useful information, and many researchers focus on the influence of topics for generating summaries with better quality [26–28]. In addition to topics, real-world entities from knowledge graphs are also meaningful knowledge that can improve the quality of summaries. Sun et al. [13] incorporate the background information of products into the popular pointer network by adding a knowledge encoder. The model performs well in the product title summarization task. Huang et al. [14] propose a knowledge graph-based abstractive model that

utilizes a dual encoder and a graph-structured encoder for summarization. However, most of them use external knowledge in supervised deep learning models. Few studies have focused on unsupervised learning models such as Textrank for document summarization [23]. Ramakanth et al. [29] propose an approach based on knowledge-based graph mining and name entity recognization for unsupervised extraction of diagnosis codes. Hou et al. [30] construct a domain knowledge base and propose an unsupervised Chinese rhetorical parsing architecture for summarization. These methods are the examples of the few studies on incorporating external knowledge into unsupervised models for summarization. However, Ramakanth et al. focus on summarization for medical records rather than news articles. Hou et al. use a rhetorical parsing architecture rather than graph models for unsupervised learning, and their model is a Chinese-oriented summarization model. In our approach, we focus on incorporating external knowledge into unsupervised graph models for extractive summarization on news articles. A knowledge-enhanced model based on unsupervised Textrank and K-means algorithms is proposed in our paper to generate better summaries.

3. Method

An unsupervised summarization model selects a subset of words or sentences from a document to generate a summary. The input of our task is a document consisting of several sentences $D = \{x_1, x_2, ..., x_n\}$, and n equals the number of sentences. The output of the task is a summary of sentences $D' = \{x'_1, x'_2, ..., x'_m\}$, where $m < n$. Usually, an unsupervised model gives scores $S = \{s_1, s_2, ..., s_n\}$ to these sentences to represent their importance, and sentences with the highest scores are chosen as the final summary.

In this section, we will describe our Knowledge Graph-Based Summarizer with K-means and TextRank (kg-KMTR) model in detail. The architecture of our model is shown in Figure 1. Kg-KMTR splits the input documents into sentences, and transfers these sentences into vectors $V = \{v_1, v_2, ..., v_n\}$ by tf-idf algorithms. An entity linking system is used in kg-KMTR to incorporate external knowledge from Wikipedia into our model. To make use of external knowledge, kg-KMTR first finds out all the entities in the sentences, and then uses another knowledge graph embedding $V' = \{v'_1, v'_2, ..., v'_n\}$ to represent them. These two kinds of sentence vectors are input into our improved Textrank algorithm, and then we have two scores $S = \{s_1, s_2, ..., s_n\}$ and $S' = \{s'_1, s'_2, ..., s'_m\}$ for each sentence. These two scores are combined as the final score for these sentences. Furthermore, K-means cluster is also used in kg-KMTR because the Textrank algorithm tends to select similar sentences as summary sentences. The top k sentences should be in different clusters to improve the diversity of our generated summaries.

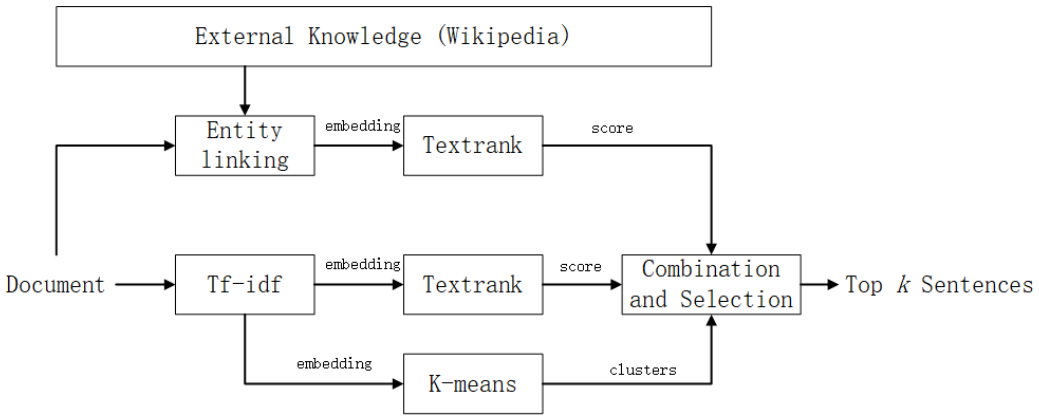

Figure 1. The overview of the Knowledge Graph-Based Summarizer with K-means and TextRank (kg-KMTR) model.

In Section 3.1, we describe how we use and improve the traditional Textrank algorithm to generate better summaries. In Section 3.2, we propose a knowledge-enhanced approach to incorporate knowledge from Wikipedia into kg-KMTR. In Section 3.3, we mainly describe our methods to combine the scores and select the sentences.

3.1. Graph-Sorting Algorithm Based on Textrank

One serious shortcoming of the traditional Textrank algorithm is that the graph is an undirected graph which do not consider the relative position between two sentences. It may affect the performance because some sentences are meaningless without the previous sentences. Therefore, we modify the original Textrank algorithm to improve the performance of our model.

3.1.1. Sentence Embedding

tf-idf algorithm is one of the most popular methods for sentence embedding in unsupervised summarization models. Our sentences are split into words, and we use this algorithm to represent the sentences of each document in our model. For each word in a document j, the tf-idf value is calculated as follows:

$$w_{ij} = tf_{ij} \times \log\left(\frac{N}{df_i}\right) \tag{1}$$

where $tf_{i,j}$ is the frequency of the word, and $\log\left(\frac{N}{df_i}\right)$ is the inverse document frequency. We get the sentence embedding by calculating the tf-idf value of each word in the sentence. Each embedding vector is filled in according to the index of the vocabulary.

Similarity between two sentences is calculated by cosine similarity where a and b are the sentence embedding vectors we get above. The value of cosine similarity ranges from 0 to 1, and the formula is shown below:

$$\text{cos\_sim}(a, b) = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} \tag{2}$$

3.1.2. Improved Graph-Sorting Algorithm

After we get the sentence embedding vectors $V = \{v_1, v_2, ..., v_n\}$, we compute a similarity matrix E by calculating the cosine similarity between sentences and normalize it. The formula of the similarity matrix is shown below:

$$E_{ij} = \frac{\text{cos\_sim}\left(v_i, v_j\right)}{\sum_{n \neq i} \text{cos\_sim}\left(v_i, v_n\right)}, \quad if \quad i \neq j \tag{3}$$

It can be seen that no directional relationship between two sentences are considered in this model. To solve this problem, we multiply $E_{i,j}$ and $E_{j,i}$ by different parameters α and $1 - \alpha$ in order to learn the importance of relative position between sentences. The parameter α ranges from 0 to 1.

$$E_{ij} = E_{ij} \times \alpha, \quad if \quad i < j \tag{4}$$

$$E_{ji} = E_{ji} \times (1 - \alpha), \quad if \quad i < j \tag{5}$$

The rest process is almost the same as the traditional graph-sorting algorithm Pagerank. After finishing this algorithm, sentences are sorted according to their own scores. The score of a sentence is computed by the following formula:

$$score(v_i) = d \sum_{v_j \in V} E_{ij} \times score(v_j) + \frac{(1 - d)}{N}. \tag{6}$$

3.2. Incorporation of External Knowledge

We believe that it is more meaningful for a summarization model to generate entities in a specific domain than any other words, so we try to make our model more knowledgeable by incorporating external knowledge from open-source knowledge graphs. We use an open-source entity linking system named Dexter (http://dexter.isti.cnr.it/) in our model to link the words in sentences to Wikipedia entries, and a knowledge graph embedding v_i' is proposed to represent sentences.

Figure 2 gives an example of entities and how entity linking works in this paper. As is shown in this figure, Michael Jeffery Jordan, United States, basketball and The National Basketball Association are the entities in knowledge graphs such as Wikipedia. Entities are the basic units of the knowledge graph, which contains rich domain knowledge such as properties and relationships. With the help of entity linking algorithms, words in a sentence are linked to corresponding entities, and we only keep these four entities to represent this sentence in this example.

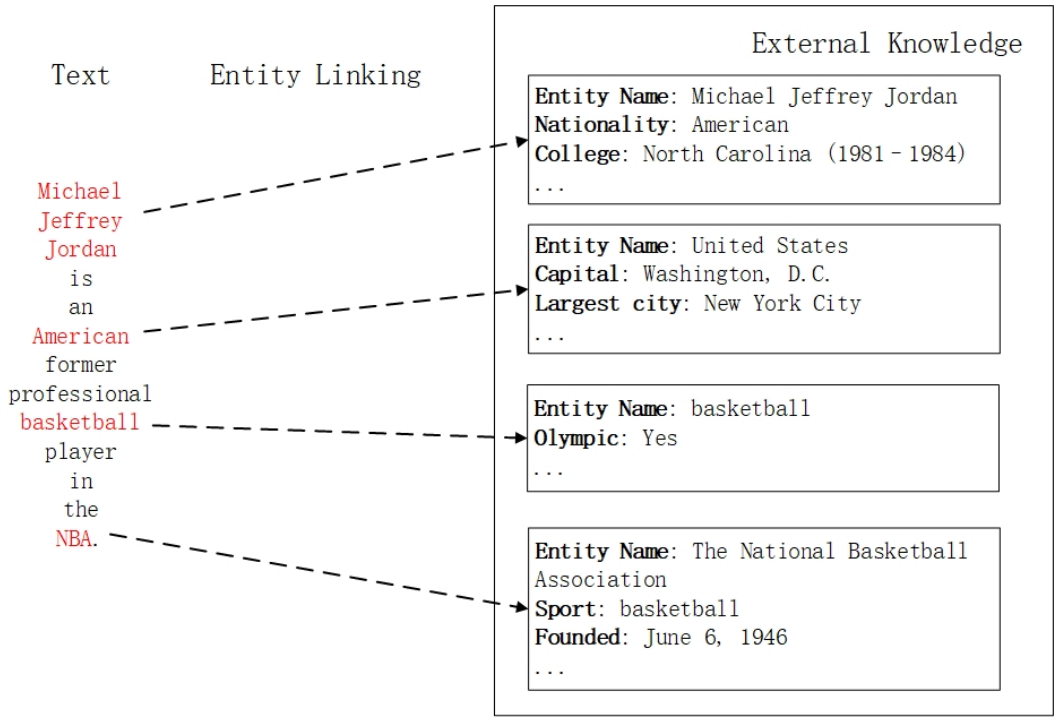

Figure 2. An example of entity linking.

After we link the words in sentences to entities in Wikipedia, a vector of entities is used $v_i' = [e_1, e_2, ..., e_n]$ to represent a sentence, and each dimension e_i is the word count of a corresponding entity. If we still use the sentence in Figure 2 as an example, the corresponding vector of this sentence is $[1, 1, 1, 1, 0, 0, ...]$ because these four entities only appear once in this sentence. The size of this vector is equal to the number of the entities in the whole document. Since a sentence corresponds to fewer entities than words, the sentence embedding vectors we get are sparse vectors. Therefore, we use word counts instead of tf-idf algorithms to represent the entities in these sentences.

We run almost the same graph-sorting algorithm for the second time to take advantage of external knowledge from Wikipedia. The only differences are the input vectors and the similarity function. The sentence embedding vectors v_i' mentioned above is used as our input vectors. Furthermore, due to the fact that these embedding vectors are very sparse, cosine similarity is not appropriate to calculate the weight of edges. If two short sentences do not contain any entities, they will have the best cosine similarity, but they actually do not have any relationships. Therefore, we use dot product instead of cosine similarity to calculate the similarity. We add the constant parameter to each dimension and then use dot product to calculate similarity between sentences. We set this constant parameter as 1 in our

experiment. As a result, a sentence with more entities tends to have a higher edge weight, and the weight is much higher if sentences have more common entities.

3.3. Combining the Scores and Selecting the Top Sentences

After finishing the algorithms described above, we get two distinct scores for each sentence. The scores from Section 3.2 are defined as $S = \{s_1, s_2, ..., s_n\}$ and the scores from Section 3.2 are defined as $S' = \{s'_1, s'_2, ..., s'_n\}$. Each score s_i from S represents the importance of words in a sentence to the entire document, and the score s'_i represents the importance of entities. To incorporate external knowledge from knowledge graphs to our model, we hope that the score of the knowledge graph can affect the original one. Therefore, linear combination is used in the kg-KMTR to get the final scores:

$$\text{score}(i) = (1 - \beta)s_i + \beta s'_i \tag{7}$$

where β is the parameter to adjust the influence of external knowledge on the model, which ranges from 0 to 1. Now we are able to sort the sentences in a document by the final score.

The last problem of our model is that the sentences with the highest scores are very similar, which may affect the diversity of the abstract. In our approach, we select the top k sentences with the help of the K-means algorithm. K-means algorithm is an popular unsupervised text clustering algorithm. The K-means algorithm for summarization represents the input sentences with vectors and classifies them into k different clusters. One centroid is defined for each cluster, and there are k centroids in total. For each turn, each sentence vector is divided into the cluster of the newest centroid, and after all these sentences are classified, the average of vectors in each cluster is defined as the new centroid. The objective function of K-means is:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| v_i^{(j)} - c_j \right\|^2 \tag{8}$$

where $v_i^{(j)}$ consists of all the sentence vectors in cluster j and c_j is the centroid in cluster j. $\left\| v_i^{(j)} - c_j \right\|^2$ is the distance between $v_i^{(j)}$ and c_j.

We use the tf-idf sentence embedding $V = \{v_1, v_2, ..., v_n\}$ as the input, and replace the original distance metric with cosine similarity mentioned in Section 3.1.1. Each sentence is classified into a cluster that ranges from 0 to k after finishing this algorithm. We select the sentence with the highest final score from each cluster, and sort these sentences according to their position in the original document. These sentences generate the final summary of our model.

4. Experiments

4.1. Data Set

We compare the performance of our model and other unsupervised models on the popular document summarization data set of The New York Times Annotated Corpus (NYT Corpus) [31]. This data set includes articles published by the New York Times between 1987 and 2007, and the summaries of these articles are written by library scientists. In our experiment, we select 2000 articles from NYT Corpus as our validation set, and 2600 articles as the test set. Since our model is an unsupervised model, training set is not needed in our experiment. The average numbers of sentences in the validation set and test set are 52.26 and 50.27, and the average sentence length of summarizes is 3.33 and 3.51.

4.2. Experiments Settings

In our experiment, we tune the hyper-parameters on the validation set and test the performance of our model on the test set. According to the performance on the validation model, we set the parameter d for the Pagerank algorithm as 0.8, and set α as 1. The parameter β is set as 0.3 to incorporate knowledge from Wikipedia into our model, and the detailed tuning process of β is shown in Section 5.1 Moreover, we use sklearn (https://scikit-learn.org/stable/) to remove the stop words and generate tf-idf sentence embeddings.

We also compare our knowledge-enhanced sentence embedding method with BERT embedding in our ablation experiment. A publicly released Albert [32] Base model (https://tfhub.dev/google/albert_base/3) is used in our experiment. The detailed parameters are described in Table 1.

Table 1. Parameters of the Albert Base model.

Parameter	Value
embedding size	128
hidden size	768
initializer range	0.02
intermediate size	3072
max position embeddings	512
attention heads	12
hidden layers	12
vocab size	30,000

4.3. Baselines and Evaluation Metrics

We compare our model with other mainstream unsupervised document summarization baselines on the test set. The baseline models are described in detail as follows.

Lead-3: A popular extractive baseline model, which selects the first three sentences as a summary. Lead-3 is a simple, but very effective algorithm.

Textrank: It is proposed by [8] to select sentences according their scores on the graph. It is one of the most common unsupervised graph-sorting algorithms to deal with text summarization tasks.

Lexrank: It is another graph-sorting algorithm proposed by [15]. Lexrank uses graph weight to compute the importance of sentences, which is similar to Textrank.

K-means: It is proposed by [18] to summarize documents by the K-means clustering algorithm. Sentences are embedded by tf-idf method in this approach.

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [33] is a commonly used automatic evaluation metric in document summarization tasks. Among the ROUGE-based metrics, ROUGE-1, ROUGE-2 and ROUGE-L are the most popular ones. They are calculated by matching the common unigrams, bigrams and the longest subsequences between the generated summary and the correct abstract respectively. The original ROUGE metrics only measure the recall rate, which leads to the fact that a longer sequence may have a higher possibility to get a better score. To deal with this problem, Rouge F1-score [34] is often used. We use ROUGE F1-score instead of recall score to measure the performance of these models in our experiment.

5. Results and Analysis

5.1. Experimental Results and Analysis

The change of ROUGE-AVG F1-score on the validation set according to the β in Equation (7) is shown in Figure 3. The ROUGE-AVG F1-score is the average of ROUGE-1, ROUGE-2 and ROUGE-L F1-score. When β is zero, external knowledge from Wikipedia is not used in our model. According to the result, when β increases from 0 to 0.3, the ROUGE F1-score increases significantly. The score decreases when β is higher than 0.3. The changes reflect the importance of incorporating external

knowledge from open-source knowledge graph to our model. In the experiments below, we set β as 0.3, and test our kg-KMTR model in our test set.

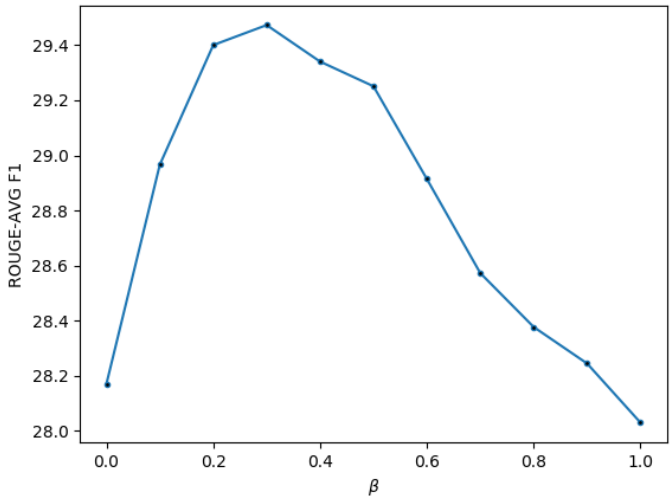

Figure 3. The change of Recall-Oriented Understudy for Gisting Evaluation (ROUGE)-AVG F1-score according to the β on the validation set.

The results of our models and the unsupervised baselines on the test set of NYT Corpus are shown in Table 2. According to the experiment, the Lead-3 model achieves a higher ROUGE score than the graph algorithms and the text clustering algorithms. The reason is that much important information of a news report is at the beginning of a document, and the other unsupervised models do not consider the relative position among sentences. Compared with the graph algorithms of Textrank and Lexrank, the K-means algorithm achieves 31.69 of ROUGE-1, 12.02 of ROUGE-2 and 29.34 of ROUGE-L, which is 2.01 higher than Textrank on average. The main difference between these two algorithms is that Textrank is more likely to select three similar sentences as the summary, which may affect the performance of the model. Document summarization method based K-means cluster solves this problem because sentences are divided into different clusters. Our kg-KMTR model achieves 36.89 of ROUGE-1, 17.66 of ROUGE-2 and 35.27 of ROUGE-L, which performs better than the other unsupervised baseline models. The average ROUGE score is 1.93 higher than the best baseline model. The reason is that kg-KMTR modifies weight of edges based on the relative position and combines the output of both Textrank and K-means to overcome the shortcomings of traditional unsupervised models. External knowledge from Wikipedia also contributes to the improvement.

Table 2. ROUGE F1-score of unsupervised models on the test set.

Model	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-AVG
Lead-3	34.64	16.22	33.18	28.01
Textrank	29.96	9.74	27.33	22.34
Lexrank	22.05	4.45	19.31	15.27
K-means	31.69	12.02	29.34	24.35
kg-KMTR (ours)	**36.89**	**17.66**	**35.27**	**29.94**

5.2. Ablation Study and Case Study

To further analyze our knowledge-enhanced method, ablation tests are performed on the kg-KMTR. In this section, we mainly concentrate on the performance of our model with and without external knowledge. We also compare our knowledge-enhanced sentence embedding method with the currently popular BERT sentence embedding method.

Our first ablation test is on the external knowledge we use in this paper. We do this experiment to show how knowledge graphs help us to generate better summaries. In this experiment, kg-KMTR is the knowledge-enhanced model proposed in our paper, and KMTR is the same model without external knowledge. The remaining parts of KMTR are the same as kg-KMTR and these two models use the same parameters. As is shown in Figure 4, knowledge enhanced kg-KMTR outperforms KMTR on ROUGE-1, ROUGE-2 and ROUGE-L. According to our experiment, the average ROUGE score increases by 1.01, which proves the effectiveness of our knowledge-enhanced approach on unsupervised document summarization.

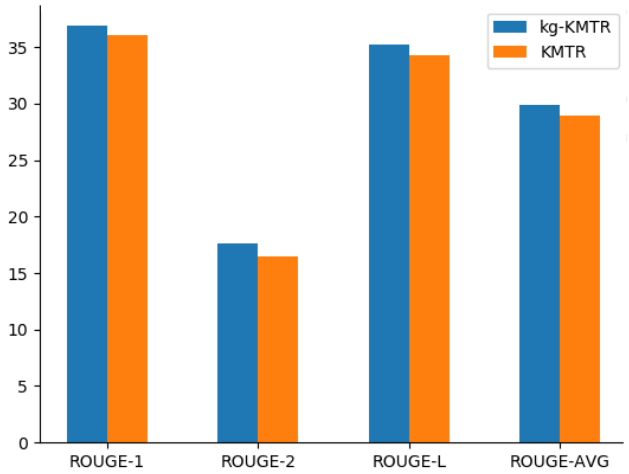

Figure 4. Performance comparison of kg-KMTR and KMTR.

We also compare our knowledge enhanced embedding approach with the popular BERT approach. We replace the tf-idf sentence embedding and the entity embedding layers with the Albert Base model described in Section 4.2 in this experiment. The [CLS] vectors are used for sentence embedding, and cosine similarity is still used as the similarity evaluation method. As is illustrated in Figure 5, simply replacing our embedding layer with the public BERT model do not improve the performance. Our embedding method based on tf-idf and entity linking is more suitable for the task.

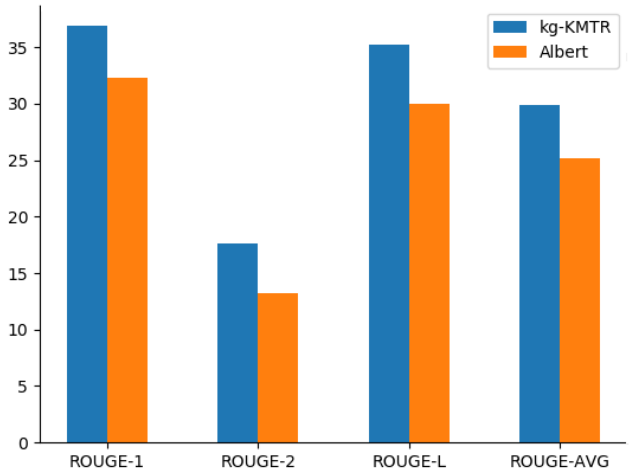

Figure 5. Performance comparison of kg-KMTR and Albert.

We also compare the generated summaries manually in order to further analyze the performance of kg-KMTR. The reference summary and the generated summaries are presented in Table 3. As is

shown in this table, the first sentence of this reference is about the bull market and the second one is of the detailed data in 2005 and 2006. It is obvious that the three sentences selected by Textrank algorithms are similar with each other. The traditional Textrank model fails to capture the topic and the important data. Our KMTR model without external knowledge overcomes this shortcoming by modifying the weight and combining the results of K-means to our sorting algorithm. It can be seen that our KMTR model picks the right data of 12.8 and the right year 2005, and the three sentences are not similar. KMTR generates a better summary than Textrank.

However, due to lack of external knowledge, KMTR fails to catch the topic sentence of the document. The last summary in Table 3 is generated by our kg-KMTR model. With the help of open-source entity linking systems, knowledge from Wikipedia is incorporated to our model. The words in red such as "bull market" and "housing market" are all the entities we discover in these sentences, and a sentence with more entities is more likely to be chosen as summary sentences in our algorithms. These entities from Wikipedia are closely related to the topic of our documents, which helps our model understand the importance of sentences better. As a result, the kg-KMTR correctly selects the topic sentence and the important data from the document and generates the best summary than the other models in our experiment.

Table 3. Examples of generated summaries on the NYT data set. (The words in red are entities found by the entity linking system).

Reference:
paul j lim article holds that four-year-old **bull market** can continue to rage on given enough **economic growth**, **corporate profits and cheap capital**. average general domestic stock fund gained **12.8 percent in 2006** , representing **surprising improvement over 2005**. investors reap rewards for risks taken during year .
Textrank:
the average diversified emerging-market stock fund , for example , soared 17.4 percent in the quarter and finished the year up 32.6 percent , according to morningstar. despite predictions that this would be the year when large-cap domestic stock funds returned to dominance , small-cap stock funds again led the domestic markets among general domestic stock funds. according to morningstar , the average small-cap growth fund advanced 11.1 percent last year , versus 7.3 percent for the average large-cap growth portfolio.
KMTR:
emboldened by an economy that proved resilient in the face of federal reserve interest-rate increases , high oil prices and a cooling housing market , the average general domestic stock fund gained **12.8 percent** last year. but it was **a surprising improvement from 2005** , when domestic stock funds gained 7.7 percent , on average , and looked to be tiring. and the average gain for equities during these years has been 18 percent.
kg-KMTR:
given enough of a dose of **economic growth , corporate profits and cheap capital** , **a bull market** that 's more than four years old – as this one is – can still rage on like a young buck. emboldened by an economy that proved resilient in the face of federal reserve interest-rate increases , high oil prices and a coolinghousing market, the average general domestic stock fund gained **12.8 percent** last year. but it was a **surprising improvement from 2005** , when domestic stock funds gained 7.7 percent , on average , and looked to be tiring.

6. Conclusions and Future Work

In this paper, we propose a knowledge-enhanced unsupervised model based on graph algorithms for extractive document summarization. External knowledge acquired by entity linking is used to incorporate external knowledge from open-source knowledge graph into our graph-based model. Compared with the traditional graph-based algorithms, the summaries generated by kg-KMTR contain more important information because of the domain knowledge learned from open-source knowledge graphs. We also improve the performance of the traditional Textrank algorithm by combining it with K-means and taking the relative position of sentences into consideration. The ROUGE evaluation on the NYT data set shows that our kg-KMTR model achieves the best result than other mainstream unsupervised models.

Although our model has achieved outstanding performance on the NYT data set, there still exist some limitations. Our kg-KMTR requires a knowledge graph that contains a large amount of domain knowledge and an entity linking algorithm with high accuracy. The quality of the knowledge graph and the entity linking algorithm may affect the performance of the model significantly. In the future, we will construct a knowledge graph in a special domain on our own and research on a new entity linking algorithm with a higher accuracy to generate better summaries.

Author Contributions: Conceptualization, T.T. and T.Y.; methodology, T.T. and T.Y.; software, T.T.; validation, X.T. and D.C.; writing—original draft preparation, T.T.; writing—review and editing, T.Y., D.C. and X.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by National Key R&D Program of China (No. 2018YFB1003801).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cheng, J.; Lapata, M. Neural summarization by extracting sentences and words. *arXiv* **2016**, arXiv:1603.07252.
2. Narayan, S.; Papasarantopoulos, N.; Cohen, S.B.; Lapata, M. Neural extractive summarization with side information. *arXiv* **2017**, arXiv:1704.04530.
3. See, A.; Liu, P.J.; Manning, C.D. Get to the point: Summarization with pointer-generator networks. *arXiv* **2017**, arXiv:1704.04368.
4. Paulus, R.; Xiong, C.; Socher, R. A deep reinforced model for abstractive summarization. *arXiv* **2017**, arXiv:1705.04304.
5. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
6. Bouscarrat, L.; Bonnefoy, A.; Peel, T.; Pereira, C. STRASS: A Light and Effective Method for Extractive Summarization Based on Sentence Embeddings. *arXiv* **2019**, arXiv:1907.07323.
7. Liu, Y. Fine-tune BERT for extractive summarization. *arXiv* **2019**, arXiv:1903.10318.
8. Mihalcea, R.; Tarau, P. Textrank: Bringing order into text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 25–26 July 2004; pp. 404–411.
9. Radev, D.R.; Jing, H.; Styś, M.; Tam, D. Centroid-based summarization of multiple documents. *Inf. Process. Manag.* **2004**, *40*, 919–938. [CrossRef]
10. Ramos, J. Using tf-idf to determine word relevance in document queries. In Proceedings of the First Instructional Conference on Machine Learning, New Jersey, NJ, USA, 21–24 August 2003; Volume 242, pp. 133–142.
11. Kiros, R.; Zhu, Y.; Salakhutdinov, R.R.; Zemel, R.; Urtasun, R.; Torralba, A.; Fidler, S. Skip-thought vectors. In Proceedings of Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 3294–3302.
12. Page, L.; Brin, S.; Motwani, R.; Winograd, T. *The PageRank Citation Ranking: Bringing Order to the Web*; Technical Report; Stanford InfoLab: Stanford, CA, USA, 1999.
13. Sun, F.; Jiang, P.; Sun, H.; Pei, C.; Ou, W.; Wang, X. Multi-source pointer network for product title summarization. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Turin, Italy, 22–26 October 2018; pp. 7–16.

14. Huang, L.; Wu, L.; Wang, L. Knowledge Graph-Augmented Abstractive Summarization with Semantic-Driven Cloze Reward. *arXiv* **2020**, arXiv:2005.01159.
15. Erkan, G.; Radev, D.R. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res.* **2004**, *22*, 457–479. [CrossRef]
16. Mallick, C.; Das, A.K.; Dutta, M.; Das, A.K.; Sarkar, A. Graph-based text summarization using modified TextRank. In *Soft Computing in Data Analytics*; Springer, Heidelberg, Germany, 2019; pp. 137–146.
17. Akter, S.; Asa, A.S.; Uddin, M.P.; Hossain, M.D.; Roy, S.K.; Afjal, M.I. An extractive text summarization technique for Bengali document (s) using K-means clustering algorithm. In Proceedings of the 2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR), Dhaka, Bangladesh, 13–14 February 2017; pp. 1–6.
18. Balabantaray, R.C.; Sarma, C.; Jha, M. Document clustering using k-means and k-medoids. *arXiv* **2015**, arXiv:1502.07938.
19. Chen, Y.C.; Bansal, M. Fast abstractive summarization with reinforce-selected sentence rewriting. *arXiv* **2018**, arXiv:1805.11080.
20. Narayan, S.; Cohen, S.B.; Lapata, M. Ranking sentences for extractive summarization with reinforcement learning. *arXiv* **2018**, arXiv:1802.08636.
21. Wang, Q.; Liu, P.; Zhu, Z.; Yin, H.; Zhang, Q.; Zhang, L. A Text Abstraction Summary Model Based on BERT Word Embedding and Reinforcement Learning. *Appl. Sci.* **2019**, *9*, 4701. [CrossRef]
22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
23. Zheng, H.; Lapata, M. Sentence centrality revisited for unsupervised summarization. *arXiv* **2019**, arXiv:1906.03508.
24. Wang, K.; Quan, X.; Wang, R. Biset: Bi-directional selective encoding with template for abstractive summarization. *arXiv* **2019**, arXiv:1906.05012.
25. Zhou, Q.; Yang, N.; Wei, F.; Tan, C.; Bao, H.; Zhou, M. Neural question generation from text: A preliminary study. In *National CCF Conference on Natural Language Processing and Chinese Computing*; Springer: Heidelberg, Germany, 2017; pp. 662–671.
26. Li, X.; Shen, Y.D.; Du, L.; Xiong, C.Y. Exploiting novelty, coverage and balance for topic-focused multi-document summarization. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, Toronto, ON, Canada, 26 October 2010; pp. 1765–1768.
27. Krishna, K.; Srinivasan, B.V. Generating topic-oriented summaries using neural attention. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018; pp. 1697–1705.
28. Kim, S.E.; Kaibalina, N.; Park, S.B. A Topical Category-Aware Neural Text Summarizer. *Appl. Sci.* **2020**, *10*, 5422. [CrossRef]
29. Kavuluru, R.; Han, S.; Harris, D. Unsupervised extraction of diagnosis codes from EMRs using knowledge-based and extractive text summarization techniques. In *Canadian Conference on Artificial Intelligence*; Springer: Heidelberg, Germany, 2013; pp. 77–88.
30. Hou, S.; Lu, R. Knowledge-guided unsupervised rhetorical parsing for text summarization. *Inf. Syst.* **2020**, *94*, 101615. [CrossRef]
31. Sandhaus, E. The new york times annotated corpus. *Linguist. Data Consort. Phila.* **2008**, *6*, e26752.
32. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. Albert: A lite bert for self-supervised learning of language representations. *arXiv* **2019**, arXiv:1909.11942.
33. Lin, C.Y. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*; Association for Computational Linguistics: Barcelona, Spain, 2004; pp. 74–81.
34. Nallapati, R.; Zhou, B.; Gulcehre, C.; Xiang, B.; dos Santos, C.N. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv* **2016**, arXiv:1602.06023.