International Journal of Knowledge *www.ijklp.org*
and Language Processing KLP International ©2016 ISSN 2191-2734
Volume 7, Number 3, **2016** **pp.43–54**

Combing Paragraph Embedding and Density Peak Sentence Clustering based Multi-Document Summarization

Baoyan Wang[1,3] and Yuexian Zou[1*] and Jian Zhang[2] and Daming Yang[4] and Yi Liu[3]
[1*]ADSPLAB/ELIP, School of ECE, Peking University, China
[2]Dongguan University of Technology, China
[3]IMSL Shenzhen Key Lab, PKU-HKUST Shenzhen Hong Kong Institute
[4]PKU Shenzhen Institute, China
*E-mail: zouyx@pkusz.edu.cn

Received May 2016; revised July 2016

ABSTRACT. *We present a novel unsupervised extractive multi-document summarization (MDS) method by combing paragraph embedding and density peak-based sentence-level clustering. Word embedding is a widely used text representation method due to its remarkable performance. However, we are aware that paragraph embedding is relatively few used in MDS. Besides, both relevance and diversity should be properly considered when generating summary. Whereas most existing MDS methods tend to quantify the degree of relevance between sentences and the other firstly, while the diversity of summary is ensured through a post-processing module. Based on these observations, three contributions are proposed in this paper. First, we compare different text representation methods for MDS thoroughly, including three classical bag-of-word methods, two word embedding methods and two paragraph embedding methods. Next, we employ density peak clustering to cluster sentences and the integrated sentence scoring method to rank them, which take relevance, diversity and length constraint into account concurrently. Third, we evaluate our method on the benchmark datasets and compare it with the other state-of-the-art methods.*

Keywords: Multi-Document News Summarization, Text Representation, Integrated Sentence Scoring Method, Density Peak Clustering

1. **Introduction.** With the explosively growing of information data overload over the Internet, consumers are flooded with all kinds of electronic documents i.e. news, tweets and blog. The end users of search engines and news providers have to read the same

information over and over again conveyed by the presence of numerous documents. There are urgent demands for multi-document summarization now more than ever, as it aims at generating a concise and informative version for the large collection of original documents that facilitates readers quickly grasp the general information of them. Most existing studies are the extractive based methods, which focus on extracting sentences directly from given materials and combining sentences together to form a summary. In this paper, we address the task of generic extractive-based summarization from the multiple documents (MDS).

Text representation has occurred as an attractive subject of research in many applications of natural language processing (NLP) due to its remarkable performance i.e. text categorization [1] , and named entity recognition [2] . Bag-of-words typically represent text as a fixed-length vector because of its concise, validness and often promising performance. Two main drawbacks are that the order and semantic of words are ignored. Word embedding [3] methods learn continuously distributed vector representations of words using neural networks, which can probe latent semantic and/or syntactic cues that can in turn be used to induce similarity measures among words. And then the paragraph is represented by averaging/concatenating the word embeddings to measure the similarity among paragraphs. Despite considering the semantic, the order of words is lost. Paragraph embedding [4] learns continuous distributed vector representations for pieces of texts, anything from a sentence to a large document. In this paper, we investigate different text representation methods for MDS thoroughly.

On the other hand, an effective summarization method always properly considers two key issues [5] : Relevance and Diversity. The extractive summarization methods can fall into two categories: supervised methods that rely on provided document-summary pairs, and unsupervised ones based upon properties derived from document clusters. The supervised methods generally treat the summarization task as a classification or regression problem [6] . For those methods, a huge amount of annotated data is required, which are costly and time-consuming. For another thing, unsupervised approaches are very enticing [7-16]. They tend to score and then rank sentences based on semantic, linguistic or statistic grouping extracted from the original documents. Whereas, most already existing methods tend to determine the relevance degree between sentences and documents firstly. And then an additional post-processing step is employed to remove redundancy and ensure the diversity of summary. They also tend to extract sentence based on greedy algorithm, which cannot guarantee the optimal summary. In this paper, we apply density peak clustering (DPC) to cluster sentences and an integrated scoring method to score sentences by considering relevance and diversity simultaneously. We extract sentences under length constraint based on dynamic programming (DP) strategy finally.

The structure of this paper is as follows. In Section 2, we briefly discuss the related work on MDS. we present the word and paragraph embedding methods utilized in our MDS algorithm in Section 3. Section 4 introduces our proposed MDS method in detail. Section 5 and Section 6 give the evaluation of the algorithms on the benchmark dataset DUC2004. In Section 7, we conclude the paper finally.

2. **Related Work.** Various extractive multi-document summarization methods have been proposed. For supervised methods, different models have been trained for the task, such as hidden Markova model, conditional random field and REGSUM [10] . Sparse coding [5] was introduced into document summarization due to its useful in image processing. Those supervised methods are based on algorithms that a large amount of labeled data is needed for precondition. The annotated data is chiefly available for documents, which are mostly relevant to the trained summarization model. Therefore, it's not necessary for the trained model to generate a satisfactory summary when documents are not parallel to the trained model. Furthermore, when consumers transform the aim of summarization or the characteristics of documents, the training data should be reconstructed and the model should be retrained necessarily.

There are also numerous methods for unsupervised extractive based summarization presented in the literature. Most of them tend to involve calculating salient scores for sentences of the original documents, ranking sentences according to the saliency score, and utilizing the top sentences with the highest scores to generate the final summary. Since clustering algorithm is the most essential unsupervised partitioning method, it is more appropriate to apply clustering algorithm for multi-document summarization. The cluster based methods tend to group sentences and then rank sentences by their saliency scores. Many methods use other algorithms combined with clustering to rank sentences. [8] [8] clustered sentences first, consulted the HITS algorithm to regard clusters as hubs and sentences as authorities and then ranked and selected salient sentences by the final gained authority scores. Wang et al. [9] translated the cluster-based summarization issue to minimizing the Kullback-Leibler divergence between the original documents and model reconstructed terms. Cai et al. [10] ranked and clustered sentences simultaneously and enhanced each other mutually. Other typical existing methods include graph-based ranking, LSA based methods, NMF based methods, submodular functions based methods, and LDA based methods. [11] used the symmetric non-negative matrix factorization (SNMF) to softly cluster sentences of documents into groups and selected salient sentences from each cluster to generate the summary. [12] employed submodular functions to address the MDS task, which take the term coverage and the textual-unit similarity into consideration. [13] evaluated the subset of summary sentences based on its projection similarity to that of the full sentences set on the top latent singular vectors. Besides, some papers considered reducing the redundancy in summary, i.e. MMR [14] . To eliminate redundancy among sentences, some systems selected the most important sentences first and calculated the similarity between previously selected ones and next candidate sentence, and add it to the summary only if it included sufficient new information.

We follow the idea of cluster-based method in this article. Different from previous work, we firstly attempt to leverage the word and paragraph embedding methods to represent text and compare with others in experiment. Besides, we propose an integrated weighted score method that can order sentences by evaluating salient scores and remove redundancy of summary at the same time. Finally, we use the dynamic programming solution for optimal salient sentences selection.

3. **The Word and Paragraph Embedding Methods.** Text representation methods map words or paragraphs into a vector space, which help learning algorithms to achieve better performance in NLP tasks i.e. text categorization. Bag-of-words methods is classical due to the concise, validness and often promising performance. However, lacking of the semantic and words' order in bag-of-words methods, it's limited of the ability to measure the similarity among sentences. Bag-of-words methods also bring about difficulty of data sparsity. In this section, we introduce the word embedding methods and paragraph embedding methods to address the problem, and compare them comprehensively.

3.1. **Neural Network Language Model (NNLM).** In [17] , the probabilistic feedforward NNLM is proposed to predict future words and generate word embedding as the by-product, which is a well-known pioneering research. Given a sequence of words, $w_1, w_2, \ldots, w_{\mathbb{C}}$, the objective function of NNLM is to maximize the log-probability:

$$\sum_{i=1}^{C} \log P(w_i \mid w_{i-n}, \ldots, w_{i-1}) \tag{1}$$

where n denotes window size of previous words, $\mathbb{C}$ is the length of the corpus, and

$$P(w_i \mid w_{i-n}, \ldots, w_{i-1}) = \frac{\exp(y(w_i))}{\sum_{k=1}^{|\mathrm{H}|} \exp(y(w_k))} \tag{2}$$

where $\mathbb{H}$ denotes the size of vocabulary, $y(w_i)$ is the possibility which w_i is the next word.

3.2. **Continuous Bag-of-words (CBOW) Model.** Similar to the feedforward NNLM, CBOW model [3] straightforward generates word embedding through context of target words instead of learning a statistical language model. For increasing efficiency, the CBOW model gets rid of hidden layers and changes from the neural network structure into the log linear structure directly. Besides, the CBOW model removes information of word order in context by using the average of word embedding, while still retains promising performance. Given a sequence of words, $w_1, w_2, \ldots, w_{\mathbb{C}}$, the objective function of CBOW is to maximize the log-probability:

$$\sum_{i=1}^{C} \log P(w_i \mid w_{i-(n-1)/2}, \ldots, w_{i-1}, w_{i+1}, \ldots, w_{i+(n-1)/2}) \tag{3}$$

$$P(w_i \mid w_{i-(n-1)/2}, \ldots, w_{i-1}, w_{i+1}, \ldots, w_{i+(n-1)/2}) = \frac{\exp(e(w_i)\overline{e(w)})}{\sum_{k=1}^{|\mathrm{H}|} \exp(e(w_k)\overline{e(w)})} \tag{4}$$

where $e(w_i)$ is the word embedding representation of word w_i, $\overline{e(w)}$ represents average of the word embedding representations of the contextual words of w_i.

3.3. **Skip-gram (SG) Model.** Contrasted with the CBOW model, the SG model [3] uses target word to predict words of context rather than predicting the nearby word based on the context. Given a sequence of words, $w_1, w_2, \ldots, w_{\mathbb{C}}$, the objective function of CBOW is to maximize the log-probability:

$$\sum_{i=1}^{C}\sum_{j=i-(n-1)/2}^{i+(n-1)/2}\log P(w_i \mid w_j, j \neq i) \tag{5}$$

$$\log P(w_i \mid w_j) = \frac{\exp(e(w_i)e(w_j))}{\sum_{k=1}^{|\mathrm{H}|}\exp(e(w_k)e(w_j))} \tag{6}$$

3.4. **Distributed Memory (DM) Model.** The DM model [4] is inspired by the method for learning word embedding. The DM model not only retains the semantics of the words, but considers the word order, at least in a small context. The DM model maps all of paragraphs into the unique vectors, and averages or concatenates the paragraph embeddings and word embeddings to predict the target word in the context. The paragraph embedding represents the missing information from the context and acts as a memory of the topic of the paragraph.

$$\sum_{i=1}^{M}\sum_{j=1}^{L_i}\log P(w_j \mid w_{j-n}, \ldots, w_{j-1}, S_i) \tag{7}$$

where M denotes the number of paragraphs in the corpus, S_i denotes the i-th paragraph, and L_i is the length of S_i.

3.5. **Distributed Bag-of-Words (DBOW) Model.** In contrast to DM model, the DBOW model [4] ignores the context words, but only predicts words randomly sampled from the paragraph. This model is also similar to the SG model in word embedding methods, and its training objective is to learn paragraph vector representations that are good at predicting the words in a small window.

$$\sum_{i=1}^{M}\sum_{j=1}^{L_i}\log P(w_j \mid S_i) \tag{8}$$

The DBOW model only needs to store the softmax weights, while the DM model has to store both softmax weights and word embeddings.

The sentence is represented by averaging the word embeddings of words appearing in that sentence or the paragraph embeddings directly in our method. Accordingly, each sentence S_i of corpus has a respective fixed-length dense vector representation.

4. **Our Proposed MDS Method.** It's universally acknowledged that a good MDS method should consider relevance and diversity properly. Whereas, most existing MDS methods, such as FGB [9] , LSA [13] , Centroid [15] , and LexRank [7] , usually concentrate on investigating the relevance degree between a sentence with the others or documents. On the other hand, those methods tend to ensure the diversity of the sentences in summary and remove redundancy by a post-processing step. Therefore, we propose a new MDS method termed as the integrated sentence scoring method, which use Density Peak Clustering (DPC) [18] to take relevance, diversity and length constraint into account simultaneously. Sentences are scored in the three aspects, and then the scores are log linearly combined. Finally, sentences are extracted to generate based on dynamic programming algorithm. Besides, our proposed method is a one-pass process and the details are given as follows.

4.1. **Relevance Score.** We show a relevance score to quantify the degree concerning how much a sentence is relevant to residual sentences in the documents. One of the underlying assumptions of DPC is that cluster centers are characterized by a higher density than their neighbors. Proceeding from it we consider that a sentence will be deemed to be more relevant and more representational when it possesses higher density meaning with more similar sentences. As the input of the DPC algorithm is the similarity matrix, the sentences are represented by different text representation methods, and then cosine equation is applied to calculate sentences' similarity. Thus, we define the function to compute the relevance scoring $SC_R(i)$ in sentences level for each sentence s_i as follows:

$$SC_R(i)=\sum_j^K f(Sim_{ij}-\omega),\quad f(x)=\begin{cases}1 & x\le 0\\ 0 & else\end{cases}, \tag{9}$$

where Sim_{ij} is the similarity value between the i-th and j-th sentence and K denotes the number of sentences in the dataset. ω represents a predefined density threshold. We set the density threshold ω following the work [18] , which excludes the sentences of lower similarity with the current sentence.

4.2. **Diversity Score.** Diversity scoring is presented to argue a good summary should not include analogical sentences. A document set usually contains one core topic and some subtopics. In addition to the most evident topic, it's also necessary to get the sub-topics most evident topic so as to better understand the whole corpus. In other words, sentences of the summary should be less overlap with one another in order to eliminate redundancy. Another hypothesis of DPC is that cluster centers also are characterized by a relatively large distance from points with higher densities, which ensure the similar sentences get larger difference scores. Therefore, by comparing with all the other sentences of the corpus, the sentence with a higher score could be extracted, which also can guarantee the diversity globally. The diversity score $SC_D(i)$ is defined in clusters level as the following function.

$$SC_D(i)=1-\max_{j:SC_R(j)>SC_R(i)} Sim_{ij} \tag{10}$$

Diversity score of the sentence with the highest density is assigned 1 conventionally.

4.3. **Length Constraint.** The longer sentence is, the more informativeness it owns. Therefore a fewer number of longer sentences tend to be extracted, which is contrary to the human summarizers who tend to produce larger number of shorter sentences. The total number of words in the summary usually is limited. The longer sentences are, the fewer ones are selected. Therefore, it is requisite to provide a length constraint. Length of sentences l_i ranges in a large scope. In this case, we should lead in taking logarithm smoothing method to handle the problem. Thus, the length constrain is defined and normalized as follows.

$$SC_L(i)=\log((\max_j L_j)/L_i) \tag{11}$$

4.4. **Integrated Sentence Scoring Method.** The ultimate goal of our method is to select those sentences with higher relevance, and better diversity under the limitation of length. In order to adapt to the integrated scoring method, $SC_R(i)$, $SC_D(i)$ and $SC_L(i)$ should be normalized by divided their own highest values firstly. We define a function comprehensively considering the above purposes as follows:

$$SC(i) = \alpha \log SC_R(i) + \beta \log SC_D(i) + \log SC_L(i) \tag{12}$$

In order to determine how to tune the parameters *α*, and *β* of the integrated sentence scoring method, we carried out a set of experiments on standard dataset.

Finally, we should generate a summary by extracting sentences under the limit of the exact length *L*. As every sentence is measure by an integrated score, the score sum of extracted sentences in summary should be as high as possible. The summary generation is regarded as the 0-1 knapsack problem:

$$\begin{aligned} &\arg\max \sum (SC(i) \times x_i) \\ &Subject\,to \sum l_i x_i \le L, x_i \in \{0,1\} \end{aligned} \tag{13}$$

The problem is NP-hard. To alleviate this problem, we utilize the dynamic programming solution [21] to select sentences until the expected length *L* of summaries is satisfied.

5. **Experimental Setup.**

5.1. **Dataset and Evaluation Metric.** The open benchmark datasets DUC2003 and DUC2004, from Document Understanding Conference, are employed in our experiments. DUC2004 consists of 50 news document sets and 10 documents related to each set. Length Limit of summary is 665 bytes. DUC2003 consists of 60 news document sets and about 10 documents for each set. The structures of both datasets are similar. Therefore, we choose DUC2003 as the development dataset for parameters tuning and DUC2004 for evaluation. There are four human generated summaries provided as ground truth for each news set. We observe that the sentences of summaries are not strictly selected in their entirety, but changed considerably.

We apply widely used ROUGE version 1.5.5 toolkit [18] to evaluate summarization performance in our experiments. Among the evaluation methods implemented in Rouge, Rouge-1 focuses on the occurrence of the same words between generated summary and annotated summary, while Rouge-2 and Rouge-SU concerns more over the readability of the generated summary. We report the mean value over all topics of the F-measure scores of these three metrics in the experiment. Note that the higher ROUGE scores, the more similar between generated summary and annotated one.

5.2. **Parameter Settings.** We set word embeddings dimensionality 50 and context size 5 empirically. We find that significant improvements using pre-trained word embeddings over randomly initialized ones. Therefore, we use Wikipedia corpus to pre-train word embeddings and fine-tune them using our corpus. We investigate how parameters *α*, *β* and the density threshold *ω* relate to our method by a set of experiments. The results of tuning parameters are shown in Figure 1-3. We find that *α*=0.8 and *β*=0.6 produce a better performance than *α*=1 and *β*=1, which indicates the effect of four scores do not equal each other for the integrated score method. Besides, the performance dropped a lot when *α* or *β* are set zero, which shows the three scores play an active role in our method. We observe that our method works best when *ω* is set 0.22.

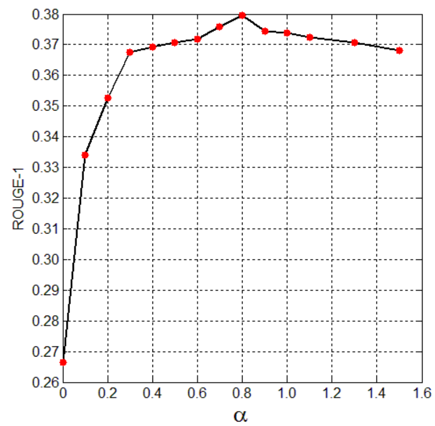

Fig.1. ROUGE-1 versus parameter α

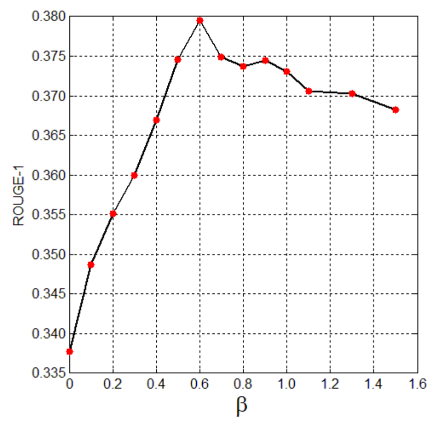

Fig.2. ROUGE-1 versus parameter β

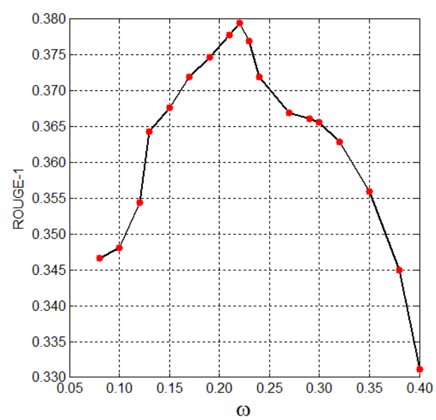

FIG.3. ROUGE-1 VERSUS DENSITY THRESHOLD Ω OF DPC

6. **Experimental Results.** We compare the word and paragraph embedding methods with different bag-of-word methods firstly: 1) BOOL (presence or absence); 2) TF (term frequency); 3) TF-ISF (combine TF with ISF). The results of these experiments are listed in Table 1.

From the Table 1, it is possible to see that BOOL term weighting achieves better results compared with TF, TF-ISF and the word embedding. It may due to the frequency of term repetition occur less in sentences. The result also indicates that paragraph embedding methods (DM/DBOW) get better results than the word embedding methods (SG/CBOW) as expected. It may be because the paragraph embedding methods take the word order into consideration and capture the semantics of sentences preferably. Besides, the DM model and CBOW model outperform the SG and DBOW respectively. It may be because the context as the input better learns the semantics than as the target when trains the word and paragraph embeddings.

TABLE 1: VALIDITY OF DIFFERENT TEXT REPRESENTATION SCHEMES IN OUR METHOD

Methods	ROUGE-1	ROUGE-2	ROUGE-SU
TF+DPC	0.38756	0.09278	0.13729
TF-ISF+DPC	0.38109	0.08934	0.13243
BOOL+DPC	0.39047	0.09559	0.13916
SG+DPC	0.37501	0.08972	0.13158
CBOW+DPC	0.38891	0.09396	0.13914
DBOW+DPC	0.39471	0.09689	0.14192
DM+DPC	**0.39947**	**0.09923**	**0.14631**

We work with the following widely used or recent published methods for general summarization as the baseline methods to compare with our proposed method. The results of these methods are listed in Table2. We divided the baseline methods into three categories:

1) DUC best
The best participating system in DUC 2004;
2) Cluster based methods
KM (Kmeans)[9] ; RTC (Rank through Cluster)[10] ; FGB (Matrix Factorization)[9] ; ClusterHITS (Cluster-based HITS Model)[8] ;
3) Others
LexRank (Graph Ranking)[16] ; LSA (Latent Semantic Analysis)[13] ; Centroid (Centroid-based Summarization)[15] ; MSSF (Submodular Function)[12] ; R2N2_ILP (Recursive Neural Networks)[6] .

For better demonstrating the results, we visually illustrate the comparison between our method and the baselines in Fig.4. From Table 2 and Fig.4, we can have the following observations: Our method clearly outperforms the DUC04 best team work on the three metrics. It is obvious that our method outperforms other rivals significantly on the ROUGE-1 metric and the ROUGE-SU metric. It can be attributed to the integrated sentence scoring method to combine paragraph embedding method with DPC, which promotes the results mutually and ensure higher quality of the summaries. Compared with other cluster based method, our method removes redundancy when clustering and considers the semantics of sentences. Our method performs slightly better than MSSF and R2N2_ILP on ROUGE-2 score. Those methods are complex and even need multiple features and postprocessor. FGB, LSA, Centroid, ClusterHITS and LexRank always need the MMR as the postprocessor to generate summary. The MMR quantifies the degree of dissimilarity between candidate sentences and already selected ones, and then select sentences based on greedy approach. The results indicate that diversity is indeed an important issue to MDS. Besides, the proposed method outperforms MMR based methods by a large margin.

TABLE 1: OVERALL PERFORMANCE COMPARISON ON DUC2004 DATASET USING ROUGE METRICS. REMARK: "–" INDICATES THAT THE METHOD DOES NOT OFFICIALLY REPORT THE RESULTS.

Methods	**ROUGE-1**	**ROUGE-2**	**ROUGE-SU**
DUC best	0.38224	0.09216	0.13233
KM	0.34872	0.06937	0.12115
RTC	0.37475	0.08973	–
ClusterHITS	0.36463	0.07632	–
FGB	0.38724	0.08115	0.12957
LSA	0.34145	0.06538	0.11946
LexRank	0.37842	0.08572	0.13097
Centroid	0.36728	0.07379	0.12511
BSTM	0.39065	0.09010	0.13218
MSFF	–	0.09897	0.13951
R2N2_ILP	0.3878	0.0986	–
DM+DPC (OURS)	**0.39947**	**0.09923**	**0.14631**

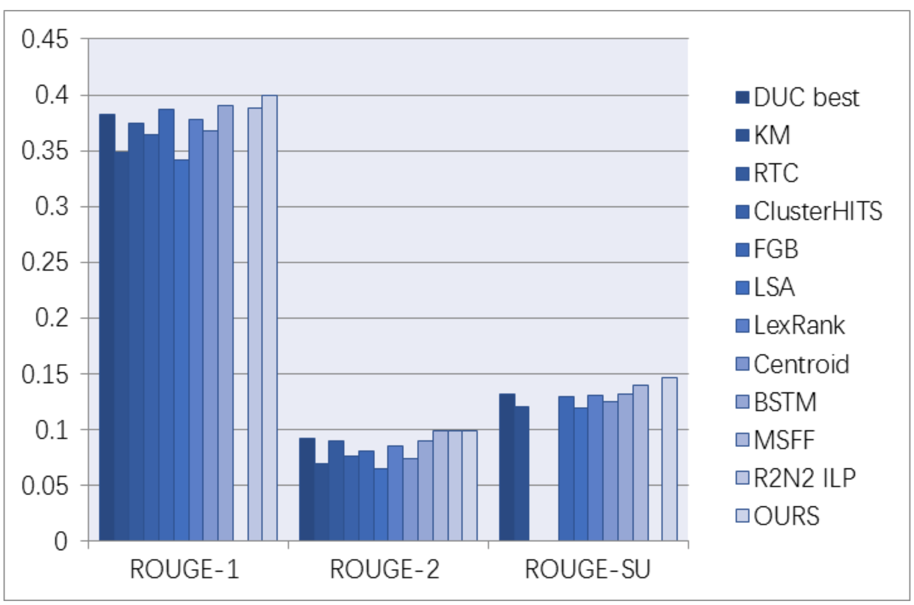

FIG.4. SUMMARIZATION RESULTS BETWEEN OURS AND OTHER STATE-OF-THE-ART METHODS.

7. **Conclusion.** In this paper, we proposed an unsupervised method to handle the task of multi-document news summarization. We applied the paragraph embedding method to represent sentence and compared with other typical bag-of-words and word embedding method comprehensively. For ranking sentences, we proposed an integrated sentence

scoring method to take relevance, diversity and length constraint into consideration. DPC was employed to measure the relevance in sentences level and diversity of sentences in clusters level at the same time. We combined those scores with a length constraint and selected sentences based dynamic programming at last. Extensive experiments on standard datasets show that our method is quite effective for MDS. In the future, we will apply our proposed method to topic-focused and updated summarization, to which the tasks of summarization have turned.

Acknowledgment. This work is partially supported by Shenzhen Science & Research projects. (No: JCYJ20150331171116474, JCYJ20150331165212372). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in Advances in neural information processing systems, 2015, pp. 649-657.

[2] A. Das, D. Ganguly, and U. Garain, "Named Entity Recognition with Word Embeddings and Wikipedia Categories for a Low-Resource Language," ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), vol. 16, p. 18, 2017.

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

[4] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in NIPS, 2013, pp. 3111-3119.

[5] H. Liu, H. Yu, and Z.-H. Deng, "Multi-Document Summarization Based on Two-Level Sparse Representation Model," in AAAI, 2015, pp. 196-202.

[6] Z. Cao, F. Wei, L. Dong, S. Li, and M. Zhou, "Ranking with Recursive Neural Networks and Its Application to Multi-Document Summarization," in AAAI, 2015, pp. 2153-2159.

[7] K. Hong and A. Nenkova, "Improving the Estimation of Word Importance for News Multi-Document Summarization," in EACL, 2014, pp. 712-721.

[8] X. Wan and J. Yang, "Multi-document summarization using cluster-based link analysis," in ACM SIGIR, 2008, pp. 299-306.

[9] D. Wang, S. Zhu, T. Li, Y. Chi, and Y. Gong, "Integrating document clustering and multidocument summarization," ACM Transactions on TKDD, vol. 5, p. 14, 2011.

[10] X. Cai and W. Li, "Ranking through clustering: An integrated approach to multi-document summarization," IEEE Transactions on Audio, Speech, and Language Processing, vol. 21, pp. 1424-1433, 2013.

[11] D. Wang, T. Li, S. Zhu, C. Ding, Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization, in ACM SIGIR, 2008, pp. 307-314.

[12] J. Li, L. Li, and T. Li, "Multi-document summarization via submodularity," Applied Intelligence, vol. 37, pp. 420-430, 2012.

[13] S. Xiong and Y. Luo, "A new approach for multi-document summarization based on latent semantic analysis," in Computational Intelligence and Design, 2014, pp. 177-180.

[14] J. Goldstein, V. Mittal, J. Carbonell, M. Kantrowitz, Multi-document summarization by sentence extraction, in NAACL, 2000, pp. 40-48.

[15] D. R. Radev, H. Jing, M. Styś, and D. Tam, "Centroid-based summarization of multiple documents," Information Processing & Management, vol. 40, pp. 919-938, 2004.

[16] 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.

[17] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," Journal of machine learning research, vol. 3, pp. 1137-1155, 2003.

[18] A. Rodriguez and A. Laio, "Machine learning. Clustering by fast search and find of density peaks," Science (New York, NY), vol. 344, pp. 1492-1496, 2014.

[19] DUC Homepage, http://duc.nist.gov/duc2004/tasks.html

[20] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in Text summarization branches out: Proceedings of the ACL-04 workshop, 2004.

[21] McDonald R. A study of global inference algorithms in multi-document summarization [J]. Advances in Information Retrieval, 2007: 557-564.