2016 Joint 8th International Conference on Soft Computing and Intelligent Systems and 2016 17th International Symposium on Advanced Intelligent Systems

Multi-document Summarization by Creating Synthetic Document Vector Based on Language Model

Dahae Kim, Jee-Hyoung Lee†
Dept. of Electrical and Computer Engineering
Sungkyunkwan University
Suwon, Republic of Korea
{kimdh35, john†}@skku.edu

Abstract— **Multi-document summarization is to create summaries covering the major information that multiple documents tell in common. For this point, the existing methods are based on hand-crafted features for word and sentence. However, it is difficult to figure out the core contents of each document with the hand-crafted features because they have the limited information presented the given documents. Moreover, there exists a limit to figure out the major information because documents with the same meaning used to be paraphrased depending on their writers. Therefore, it is necessary to represent the semantic meanings of documents as well as sentences through understanding natural language. In this paper, we propose a new multi-document summarization system by creating a synthetic document vector covering the whole documents based on Language Model, whose is well-known for learning the semantic features in text. We experimented with DUC 2004 dataset provided by Document Understanding Conference (DUC) and the results show that our method summarizes multiple documents effectively based on their core contents.**

Keywords-Multi-document summarization; Core content; Major Information; Synthetic document vector; Language model;

I. INTRODUCTION

Many documents are created in large quantities for a day through e-mailing, new articles published online, and any public/private purposes. With the growing number of the documents, the information overload is caused and calls for methods to retrieve the desired information automatically. The main goal of these summarization systems is to deliver the major information that the multiple documents tell in common. The summarized information helps people to better understand the large amount of documents in a short time [1, 2]. One major issue for the multi-document summarization is to capture the core contents of documents. In order to consider this point, the existing summarization methods are based on hand-crafted features, such as word frequencies, position and length of sentence, or use additional software implemented for natural language processing like Named Entity Recognizer, WordNet. Cao [3] created 23 features about words and sentences to model vectors at the various view. Based on the features they scored each sentence through learning recursive neural network. Yan [4] proposed a graph-based sentence ranking algorithm using Semantic Role Labeling (SRL). The making use of SRL could be helpful for capturing the semantic information within sentences. However, it is difficult to figure out the core contents of each document with the hand-crafted features because they have the limited information presented the given documents without understanding the contents in documents semantically.

Additionally, many existing methods are based on Bag-of-words vector space for word or sentence modeling [5]. The Bag-of-words (BOW) is one of the vector representation methods that has been widely used. It represents vectors with appearance of words from a dictionary. However, BOW based models are incapable of modeling complicated representation in natural language because this kind of approaches cannot consider the semantic or grammatical features. In order to consider this point, Zhang [6] proposed a summarization method based on distributed word representation. A sentence vector is represented by summation of word vectors in a context. However, there are many sentences with the same meanings and they should be paraphrased easily depending on writers. In order to consider these problems, we focus on understanding sentences or documents for capturing the core contents of documents. In this paper, we propose a new multi-document summarization system by creating a synthetic document vector covering the whole documents based on Language Model.

The rest of this paper is organized as follows. Section 2 introduces Distributed representation and Paragraph Vector. Section 3 presents a detailed description of our summarization method. Section 4 compares the proposed method to baselines. Finally, we conclude our works in Section 5.

II. BACKGROUND

In this section, Distributed representation of words and Paragraph Vector (PV), which is one of Language models, are described as background of this paper.

A. Distributed representation

Distributed representation of words is to learn the semantic and syntactic feature in words. Words can be represented as a dense vector with a prediction task which estimates the conditional probability of a word given the other words in a context. This probabilistic models was firstly proposed by [7] and many language models have been researched [8, 9].

978-1-5090-2678-4/16 $31.00 © 2016 IEEE
DOI 10.1109/SCIS&ISIS.2016.159

The objective of the distributed representation is to maximize log probability of the training data. Loss function L is defined as follows.

$$L = \frac{1}{T}\sum_{t=c}^{T-c} \log P(w_t|w_{t-c}, \dots w_{t+c})$$

The prediction task is typically done via a multiclass classifier, such as softmax. There, we have

$$P(w_t|w_{t-c}, \dots w_{t+c}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}}$$

Each of y_i is un-normalized log-probability for each output word i, computed as

$$y = b + Uh(w_{t-c}, \dots w_{t+c}; W)$$

where U, b are the softmax parameters. h is constructed by a concatenation or average of word vectors $w_{t-c}, \dots w_{t+c}$ in a context. After training the model, words with similar meaning are mapped to a similar position in the vector space. These representations have been proven very effective in natural language processing tasks such as paraphrase detection [10], question answering system [11] and text summarization for this paper.

B. Pragraph Vector: Distributed Memory

Paragraph Vector (PV) is a language model proposed by [8]. The model focus on capturing the semantic meanings in the variable size of text by inserting a memory vector to the standard language model.

In Fig. 1, the concatenation or average of vector with a context of words is used to predict the next word. The prediction task changes the word vectors and the paragraph vector. Each paragraph vector represents the missing information from the current context and can be used as a memory of the topic of the paragraph.

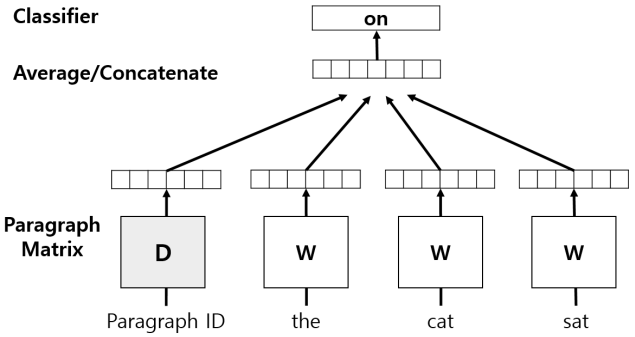

Figure 1. Paragraph Vector

One of important advantages of paragraph vectors is that they are learned from unlabeled data and also compensate some of the key weaknesses of bag-of-words models as they inherit semantics properties of the words. Another advantage of Paragraph Vector is that they take into consideration the word order. Therefore, we adopt PV to sentence and document vector modeling for our summarization system.

III. PROPOSED METHOD

We begin the description of the full framework with the following three sub sections: Sentence vector representations, Projecting documents into semantic sentence spaces, Sentence extraction using synthetic document vector. Fig. 2 shows the overview of our proposed method.

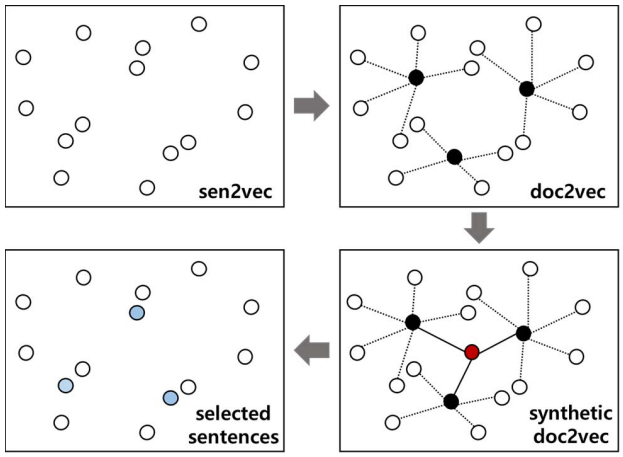

Figure 2. Overview of proposed method

We first map all sentences in document sets into a semantic vector space through Paragraph Vector. Then, we project each document into the semantic vector space. After projecting documents, we create a synthetic document vector which contains the overall meanings of the given document set. Based on the created synthetic document vector, we extract important sentences containing the major information of the document set.

A. Sentence Vector Representations

It is necessary to model sentence vectors containing their semantic and syntactic meanings for understanding document contents. In this paper, we employ Paragraph Vector to represent sentence vectors. For the extractive summarization task, we train the model for sentence-level representations.

As described in Section 2, all the sentences in the document sets are mapped to a unique vector, represented by a column in matrix D_s instead of Paragraph Matrix in Fig. 1 and other settings are the same with the original model. First of all, we train to get word vectors W, softmax weghts U, b and sentence vector D_s on sentences in document sets.

B. Projecting Documents into Semantic Sentence Spaces

Since we focus on understanding the core contents of a document, we project each document into the semantic space that sentences are distributed. Projected document vector into semantic sentence space is located in the most similar sentence to oneself then we can figure out which sentence tells core contents at the document view.

For document vector representation, we apply the trained PV model in a different way. The inference stage of PV is used to get document vectors D_d for all the documents while holding W, U, b fixed. A document consist of the sequence of

all words $d = \{w_1, w_2, \dots w_n\}$ and it is inferred through some changes of original equations. As shown in (1), the objective function L' to maximize the average log probability is defined as below.

$$L' = \frac{1}{T}\sum_{t=c}^{T-c} \log P(w_t | w_{t-c}, \dots w_{t+c}, d) \quad (1)$$

$$P(w_t | w_{t-c}, \dots w_{t+c}, d) = \frac{e^{\tilde{y}_{w_t}}}{\sum_i e^{\tilde{y}_i}} \quad (2)$$

$$\tilde{y} = b + Uh(w_{t-c}, \dots w_{t+c}, d; W) \quad (3)$$

C. Sentence Extraction using Synthetic Document Vector

In this section, we present how to extract important sentences from a document set using Synthetic document vector. In this paper, the salience of sentences is determined by calculating similarity to Synthetic document vector containing the major information of whole documents. Synthetic document vector is simply created by averaging individual document vectors. For example, the red point as shown in Fig. 3 represents the created synthetic document vector (Synthetic doc2vec) for the given document set containing three documents.

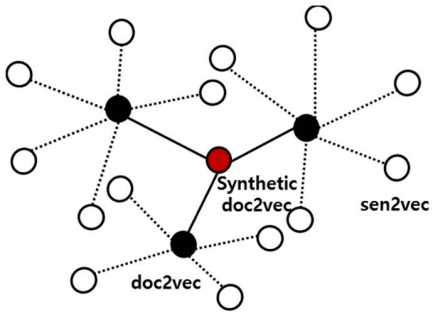

Figure 3. Synthetic document vector creation

The score of a sentence s_i is calculated by cosine similarity to the created synthetic document vector as shown in (4).

$$Score(s_i) = cosim(s_i, d_{syn}) = \frac{\overrightarrow{s_i} \cdot \overrightarrow{d_{syn}}}{\|s_i\| \|d_{syn}\|} \quad (4)$$

$$RedundancyCheck(s_i, summary) = \frac{\sum_{s_j \in summary} cosim(s_i, s_j)}{len(summary)} \quad (5)$$

We rank sentences based on their score $Score(\cdot)$ and extract sentences sequentially using the rank. In addition to this, we design a term to select non-redundant sentences effectively. $RedundancyCheck(s_i, summary)$ as shown in (5) checks the similarity of s_i to the already extracted sentences into $summary$ before we extract the sentence.

procedure SE;
inputs:
  D: doc2vec set (d_1, d_2, ⋯, d_{N_d}) in a document set
  S: sen2vec set (s_1, s_2, ⋯, s_{N_s}) in D
  α: summary length limit
  δ: threshold for redundancy check
outputs: summary for a document set
$\boldsymbol{d_{syn}} := \sum_{d \in D} d/|D|$;
  for i := 0 **to** N_s **do**
    $Score(s_i) := cosim(\boldsymbol{s_i}, d_{syn})$;
  end {for}
summary := $\underset{s \in S}{\operatorname{argmax}}\, Score(s)$;
while *len(summary)* < α **do**
  Let $\boldsymbol{S}$ be a set of sen2vec except for extracted sentences
  $\boldsymbol{s}$: = $\underset{s \in S}{\operatorname{argmax}}\, Score(s)$;
  if *RedundancyCheck*($\boldsymbol{s}$, *summary*) < δ **then**
    summary += $\boldsymbol{s}$;
end {while};
end {SE};

Figure 4. Pseudo-code of sentence extraction algorithm

As shown in Fig. 4, the sentence extraction algorithm (SE) takes a set of document vectors (doc2vec), a set of sentence vectors (sen2vec), a length limit parameter α, and a threshold δ for redundancy check. Initially, sentence of the first rank is extracted into the summary. The algorithm incrementally adds sentence s until the length of summary is limited by α. In every step, we extract a sentence with the highest score in S $\underset{s \in S}{\operatorname{argmax}}\, Score(s)$ and then exclude the sentence from S at the next loop. Moreover, we check the redundancy of sentences in the summary in order to make summaries informative.

IV. Experiment

A. Datasets

The most commonly used evaluation corpora for summarization are the ones published by the Document Understanding Conference [1] (DUC). We use DUC 2004 dataset for our experiment. This dataset consists of 50 clusters of related documents, each of which contains 10 documents. Each cluster of documents also includes four gold standard summaries used for evaluation. As in the DUC 2004 competition, we allowed 665 bytes (about 100 words) for each summary including spaces and punctuation.

[1] http://www-nlpir.nist.gov/projects/duc/intro.htm

B. Settings

For the training of Language model, we set the embedding parameter I into 300 dimensions. The dimensions of document and sentence are the same for calculating their similarity. In (4), the parameter δ is set to 0.9 practically.

C. Baselines

We describe several state-of-the-art systems and three systems implemented based on different vector modeling. For implementation of BOW, Doc2vec, Doc2vec+TextRank, the (5) and algorithm SE in Fig. 4 are applied.

- Peer65 [12]: This system was the best among those that entered the official DUC 2004 evaluation. It employs a Hidden Markov Model, using topic signature as the only feature. The probability of one sentence being selected in the summary also depends on the importance assigned to its adjacent sentences in the input document.
- Submodular [13]: The advantage of using a submodular function to estimate summary importance is that there is an efficient algorithm for incrementally computing the importance of a summary with a performance guarantee on how close the approximate solution will be to the globally optimal one.
- RegSum [14]: RegSum employs a supervised model for predicting word importance. RegSum combines the weights estimated from three unsupervised approaches, along with features including locations, part-of-speech, name-entity-tags, topic categories and contexts.
- BOW: It is implemented based on the bag-of-words representation. There exist 17,636 words in the input document sets and each word appeared in the sentence is encoded as 1, otherwise, 0.
- Doc2vec: It is implemented based on the document vector described in sub section B of Section 3. The dimension of each document set into 300.
- Doc2vec+TextRank: TextRank is one of the most popular algorithm for summarization task proposed by [5]. The input text is represented as a graph $G(V, E)$, where V is the set of sentences in the input. The cosine similarity between sentences is used as weight. We apply Doc2vec to vertices in TextRank.

D. Experimnet Results

We evaluated experiments using ROUGE measure [15] which is widely used in evaluation for summarization. ROUGE-N is an n-gram recall between a candidate summary c and a set of reference summaries R. ROUGE-N is calculated as below (6).

$$ROUGE - N = \frac{\sum_{c \epsilon R} \sum_{gram_n \epsilon C} Count_{match}(gram_n)}{\sum_{c \epsilon R} \sum_{gram_n \epsilon C} Count(gram_n)} \quad (6)$$

Where n denotes n-gram and $Count_{match}(gram_n)$ is the total number of n-grams that occurring in the candidate summary and reference summaries. $Count(gram_n)$ is the total number of n-grams that occurring in reference summaries. C is the candidate summary and R is the human summaries set referred to reference summaries.

The TABLE I shows the recall and F-measures evaluated by ROUGE-1 and ROUGE-2 measures. We averaged scores of all the document sets in DUC 2004. We referred to the score of Peer 65, Submodular and RegSum written by [16].

TABLE I. Experiment Results on DUC 2004

System	ROUGE-1		ROUGE-2	
	Average Recall	*Average F-Measure*	*Average Recall*	*Average F-Measure*
Peer 65	-	0.3794	-	0.0896
Submodular	-	0.3918	-	0.0935
RegSum	-	0.3857	-	0.0975
BOW	0.3814	0.3528	0.0856	0.0795
Doc2vec	0.3981	0.3775	0.1048	0.0987
Doc2vec+TextRank	0.4051	0.3780	0.1064	0.0986
Synthetic Doc2vec (Proposed Method)	**0.4290**	**0.3994**	**0.1245**	**0.1154**

The experiment results show that our method outperforms the state-of-the-art systems. As you can see in TABLE I, the ROUGE-2 score considering bigrams is much higher than other methods in our method.

We conducted another experiment to check how similar the contents of our summaries are with the human summaries in terms of semantic view. After we create summaries, we projected summaries into the semantic sentence space and then calculated cosine similarity between embedded summary vectors (Summary2vec). The Fig. 5 below shows the comparison of Summary2vec similarity between human summaries and system summaries.

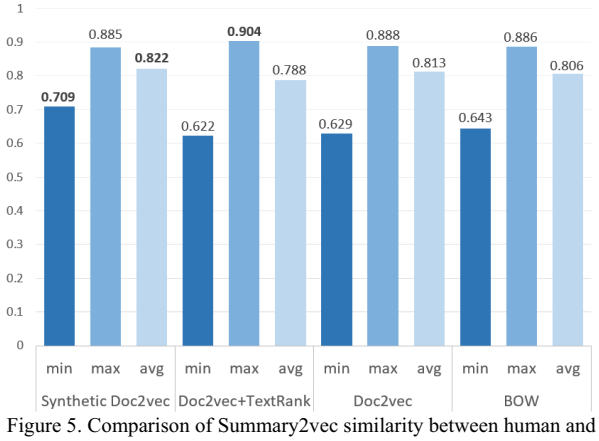

Figure 5. Comparison of Summary2vec similarity between human and systems summaries

We obtained the minimum, maximum, average scores of the similarity about DUC 2004. As shown in Fig. 5, although the maximum score of Doc2vec+TextRank is the highest, the minimum score in Synthetic Doc2vec is higher than other methods as much of a difference. It means that our model

guarantees general performance for all document sets as well as high quality.

V. CONCLUSION

In this paper, we proposed a novel multi-document summarization method by creating a synthetic document vector based on the Paragraph Vector model. Through projecting documents into the semantic sentence spaces and creating the synthetic document vector, we could capture the major information in multiple documents. The proposed methods were compared with state-of-the-art approaches. The experiment result This research was supported by ICT R&D program of MSIP/IITP (10041244, Smart TV 2.0 Software Platform).s shows that the proposed method is outstanding to figure out the core contents from related documents.

In the future work, we are going to research the abstractive summarization task that requires more understanding about natural language.

ACKNOWLEDGMENT

This research was supported by ICT R&D program of MSIP/IITP (10041244, Smart TV 2.0 Software Platform). And This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (B0101-16-0559)

REFERENCES

[1] Dahae Kim and Jee-Hyoung Lee, "Topic-focused Multi-Document Summarization Based on Genetic Algorithm," The 16th International Symposium on advanced Intelligent Systems, pp. 733-740, 2015.

[2] Noo-ri Kim, Hana Cho, and Jee-Hyong Lee, "Document Summarization Considering Relative Characteristics in a Document Set," The 10th Asia Pacific International Conference on Information Science and Technology, pp. 121-124, 2015.

[3] Ziqiang Cao, Furu Wei, Li Dong, and et al., "Ranking with Recursive Neural Networks and Its Application to Multi-Document Summarization," Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 2153-2159, 2015.

[4] Su Yan and Xiaojun Wan, "SRRank: Leveraging Semantic Roles for Extractive Multi-Document Summarization," IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE, vol. 22, no. 12, pp. 2048-2058, 2014.

[5] Rada Mihalcea, and Paul Tarau, "TextRank: Bringing order into texts," Association for Computational Linguistics, 2004.

[6] Yong Zhang, Meng Joo Er, and Rui Zhao, "Multi-Document Extractive Summarization Using Window-Based Sentence Representation," Computational Intelligence, pp. 404-410, 2015.

[7] Yoshua Bengio, Rejean Ducharme, and et al., "Neural probabilistic language models," The Journal of Machine Learning Research, vol. 3, pp. 1137-1155, 2003.

[8] Quoc Le, and Tomas Mikolov, "Distributed Representations of Sentences and Documents," International Conference on Machine Learning, vol.32, pp. 1188-1196, 2014.

[9] Eric H. Huang, Richard Socher, Christopher Manning, and Andrew Ng., "Improving word representations via global context and multiple word prototypes," In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, vol.1, pp. 873-882, 2012.

[10] Richard Socher, Eric H. Huang, Jeffrey Pennington, and et al., "Dynamic pooling and unfolding recursive autoencoders for paraphrase detection," In Advances in Neural Information Processing Systems, 2011.

[11] Jason Weston, Sumit Chopra, and Antoine Bordes, "Memory networks," In *International Conference on Learning Representations* (ICLR), 2015.

[12] John M. Conroy, Jade Goldstein, Judith D. Schlesinger, and Dianne P. O'leary, "Left-brain/right-brain multi-document summarization," In Proceedings of the Document Understanding Conference, 2004.

[13] Hui Lin, and Bilmes, Jeff, "A class of submodular functions for document summarization," Proceedings of ACL, pp. 510-520, 2011.

[14] Kai Hong, and Ani Nenkova, "Improving the estimation of word importance for news multi-document summarization," In Proceedings of EACL, Gothenburg, Sweden, April, 2014.

[15] Chin-Yew Lin, "Rouge: A package for automatic evaluation of summaries," the ACL-04 Workshop, pp.74-81, 2004.

[16] Kai Hong, John M. Conroy, and et al., "A Repository of State of the Art and Competitive Baseline Summaries for Generic News Summarization," Ninth International Conference on Language Resources and Evaluation (LREC), pp. 1608-1616, 2014.