A Text Document Clustering Method Based on Weighted BERT Model

Yutong Li[1] , Juanjuan Cai[2] , Jingling Wang[1*]
1. School of Information and Communication Engineering, Communication University of China
2. Key Laboratory of Media Audio &Video (Communication University of China), Ministry of Education
Beijing, China
{yutong_li,caijuanjuan,wjl}@cuc.edu.cn

Abstract—**Traditional text document clustering methods represent documents with uncontextualized word embeddings and vector space model, which neglect the polysemy and the semantic relation between words. This paper presents a novel text document clustering method to deal with these problems. Firstly, pre-trained language representation model Bidirectional Encoder Representations from Transformers (BERT) is utilized to generate sentence embeddings. Then, two sentence-level weighting schemes based on named entity are designed to enhance the performance. Finally, the k-means clustering algorithm is applied to find groups of similar documents. Experimental results on four datasets indicate that the proposed weighted method achieves higher accuracy than unweighted average method. Friedman tests conducted separately with F1 score and Adjusted Rand Index (ARI) values both validate better overall performance of our proposed method.**

Keywords—text mining; document clustering; transformer; named entity

I. Introduction

The text document clustering has become a high-profile technique in the field of data mining recently. It has a wide range of applications such as in document organization, topic detection and information retrieval. The text document clustering is the task of partitioning the document collection into disparate and meaningful groups. Its main target focuses on dividing the text documents into separate clusters so that the text documents belonging to one cluster are much similar to each other and different from documents in other clusters [1].

Clustering a large collection of documents is usually a complex process. One reason for this is that text documents in unstructured form cannot be simply processed by clustering algorithms, which require the input to be a fixed-length feature vector [2,3]. Therefore, it is critical to firstly figure out how we represent the text documents in the same and structured way.

One popular document representation approach is the Vector Space Model (VSM). It converts unstructured text documents into a high dimensional vector where every term represents an attribute. Despite the popularity, this model has two major weaknesses, i.e., it loses the ordering of the words and fails to capture semantic relation between the words [4]. Thus, different sentences can have exactly the same representation, as long as the same words are used.

Benefited from the rapid development of natural language processing (NLP), many neural network language models (NNLM) have been used to address representation problem in text document clustering. Word embeddings trained by Word2vec [5] and Glove [6] are commonly applied as basic building blocks for text representation. Documents can be expressed by these word embeddings with weights. However, these word embeddings are uncontextualized and neglect the polysemy of words.

In addition to the text representation problem, the design of weighting scheme is another problem worthy of concerning. The role of a weighting scheme is to further organize the text representation according to the amount of information this text provides in the document [1]. Most weighting methods in text clustering are based on term features such as TF (Term Frequency) and TF-IDF (Term Frequency and Inverse Document Frequency), which only emphasize the significance of each word in the documents, rather than that of each sentence or each paragraph.

To deal with the issues mentioned above, we propose a text document clustering method based on BERT (Bidirectional Encoder Representations from Transformers) model [7]. This language representation model makes use of a tremendously huge amount of plain text data from the BooksCorpus and English Wikipedia and is trained in an unsupervised way. In our method, BERT model is utilized in the embedding module to generate contextualized sentence embeddings and two sentence-level weighting schemes based on named entity are designed in the weighting module to improve the document representation. The rest of the paper is set out as follows. Section II introduces the proposed method. Section III shows the results of the experiments and relevant discussion. Finally, conclusion of the work and future work are offered in section IV.

II. Proposed Method

As illustrated in Fig. 1, our document clustering method is divided into three modules, i.e., (1) the embedding module dealing with text document representation problems, (2) the weighting module resolving weighting scheme problems and (3) the clustering module aggregating groups of similar documents. The following subsections explain the details of these modules.

A. Embedding Module

In the embedding module, the pre-trained language representation model BERT is used to generate contextualized sentence embeddings in the first step.

978-1-7281-4390-3/20/$31.00 ©2020 IEEE

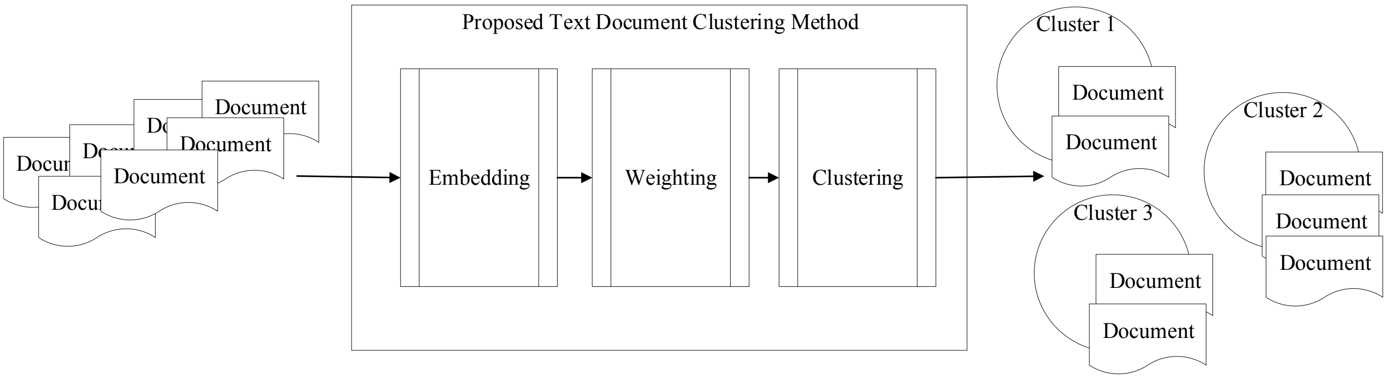

Fig. 1. The framework of proposed text document clustering method.

We choose BERT base-sized model structure, which is a 12-layer bidirectional Transformer encoder consisting of the original implementation explained in [8]. As BERT model is mainly based on the encoder structure of sequence model Transformer, for the sake of descriptive integrality, we have to briefly introduce the Transformer and describe the details of the encoder stacking in the encoding component of transformer at first. The Transformer is a network architecture based solely on attention mechanisms, which eliminates the use of convolutions and recurrence completely [8]. Self-attention, the core idea the Transformer is built on, is an attention mechanism that relates different positions of a single sequence for purpose of calculating a representation of the sequence. Fig. 2 illustrates that an encoder contains two main sub-layers, i.e., (1) a multi-head self-attention layer computing self-attention for each position of the input vector and (2) a fully connected feed-forward neural network independently and equivalently applied to every position. On each of these layers, residual connection and layer normalization are both employed.

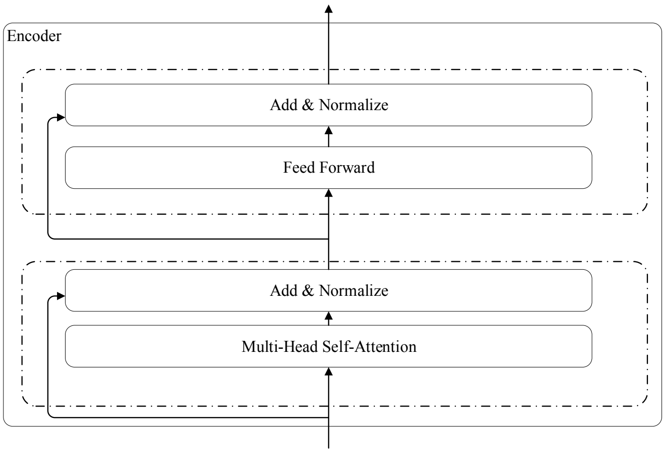

Fig. 2. Details of the encoder.

As demonstrated in Fig. 3, twelve encoders are stacked on the top of each other in BERT base-sized model. The special classification token ([CLS]) is defined as the first token of every sequence. BERT uses token-level embeddings called WordPiece from [9] as its token embeddings. For each token in a sequence, its input representation is the sum of the corresponding token, segment, and position embeddings.

In our embedding module, rather than the last hidden layer, the output of the second-to-last hidden layer (i.e. eleventh encoder) is employed as the embeddings of all of the tokens in the sentence. That is mainly because we apply the pre-trained BERT to our own method without fine-tuning, the output of the last layer could be too close to BERT's own target function (i.e. next sentence prediction and masked language model).

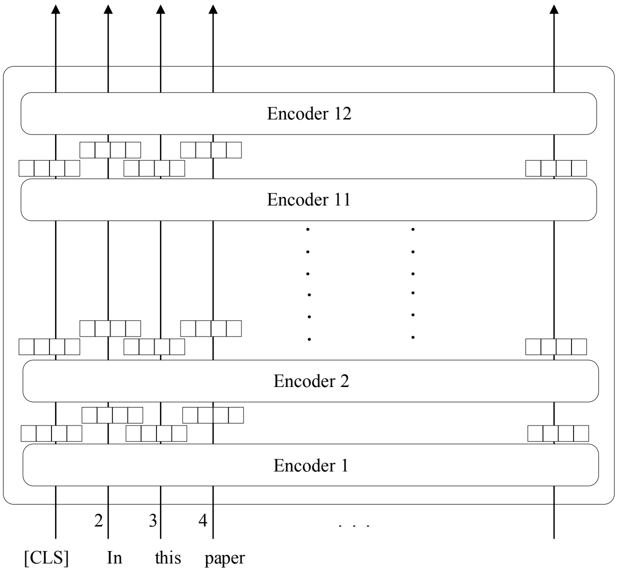

Fig. 3. The model structure of base-sized BERT.

For each sentence with l tokens, a $l \times 768$ dimension embedding is generated, 768 is the hidden size of base-sized BERT model. Then, average pooling strategy is utilized on the embedding to reduce the dimension l to 1, which conveys the idea that the meaning of a sentence is represented by all words contained in it. Generally, this module maps each variable-length sentence in text documents to a 768-dimensional fixed-length sentence embedding.

B. Weighting Module

In this module, we have designed two weighting schemes based on named entity. Anything that can be referred to with an appropriate name is a named entity, e.g., a person, an object, or an institution. Moreover, this term is commonly extended to contain things that are not entities per se, including temporal expressions such as dates and times, and even numerical expressions like prices [10]. Named entities in text documents often include important factoids, which are more informative than other words in sentences. Since sentences containing more information words are more valuable, we pick the number of named entities of particular types that each sentence contains as the sentence-level feature to design our weighting methods on sentences. It is worth noting that the types of named entities are best chosen carefully and vary from task to task. For a specific text document clustering task, the appropriate types of named entities are selected with good representativeness and differentiation according to the main content of the document collection. The detailed description of the two weighting schemes is as follows.

1) The first weighting scheme

A simple WA (Weighted Average) approach is applied to the first weighting scheme. For every sentence s in document d, the sentence weight is

$$w_s = n_s + 1 \tag{1}$$

Where n_s denotes the number of named entities of particular types in sentence s. Considering that there are sentences without named entities, the minimum of this weight is 1,which ensures that all sentences participate in calculation. Then we can get the representation of document d as

$$v_d = \frac{\sum_{s \in d} w_s \times v_s}{\sum w_s} \tag{2}$$

This WA scheme highlights the difference in the amount of information between all the sentences contained in each document per se. It is simple and easy to extend. When new documents are added to the document collection, only the representation of new documents needs to be calculated, instead of updating the representation of all documents in the collection.

2) The second weighting scheme

The second scheme uses WR (Weighted Removal) approach directly inspired by [11]. In the weighting step, each sentence s in document d is weighted according to the following formula:

$$w_s = \frac{a}{a - p(s)} \tag{3}$$

In formula (3), $p(s)$ is the probability of sentence s defined as

$$p(s) = \frac{n_s}{N} \tag{4}$$

Where N is the total number of named entities of particular types that the whole document collection D contains. And a is a parameter proportional to the maximum of $p(s)$.

$$a = \alpha p_{\max}(s) \tag{5}$$

In this paper, a is fixed to 10, which puts w_s in $[1,1.11]$. Note that for more informative sentence s, the weight w_s is bigger, so this naturally leads to an up weighting of the sentence. Then we can get the representation of document d as

$$v_d = \tfrac{1}{|s|}\sum_{s \in d} w_s \times v_s \tag{6}$$

For each document d in the collection D, the first principle component is removed to obtain the final document representation in the removing step. D is a matrix whose columns are $\{v_d : d \in D\}$, and let u be its first singular vector. For all documents in collection D, correction on their representation is provided by the following formula:

$$v_d = v_d - uu^T v_d \tag{7}$$

In this WR scheme, the weight for each sentence in first step not only reflects the importance of the sentence, but also associates the document to which the sentence belongs with the entire document collection. The removal of the first principle component in second step partly reduces the commonality between all documents, which makes the differences between all documents more prominent.

In summary, the sentence embeddings are organized by two weighting schemes in this module for the purpose of representing the documents. Each variable-length document is represented by a 768-demensional fixed-length vector on the basis of weighted 768-demensional sentence embeddings.

C. Clustering Module

In the last clustering module, we feed document vectors to the k-means clustering algorithm. It is a common partitioning algorithm which is widely used in text clustering. K-means text clustering algorithm divides n text documents into k clusters. k is defined in advance. Firstly, k documents are randomly chosen as initial centroids. Each document is assigned to the nearest centroid with distance or similarity measure and the relevant documents belonging to the same centroid are gathered into a cluster. Then new cluster centroids are calculated, and documents are rearranged. The measure value between each text document and cluster centroids iteratively updates the cluster centroids and reorganizes the clusters until the termination condition is met or there is no change in clusters [12].

III. Experimental Results

In order to compare the performance of methods separately using UA (Unweighted Average) sentence embeddings, weighted sentence embeddings with first WA scheme and weighted sentence embeddings with second WR scheme, several experiments are conducted in this section.

A. Datasets and Evalution Metrics

We test our method on text documents from Reuters-21578, Distribution 1.0 in Natural Language Toolkit (NLTK). Considering that this corpus is a collection of the Reuters financial news, six specific types of named entities are counted as shown in Table I:

TABLE I. THE TYPES OF NAMED ENTITIES COUNTED IN THE EXPERIMENTS

Index	**Type**	**Description**
1	PERSON	People, including fictional
2	NORP	Nationalities or religious groups
3	ORG	Companies, agencies, institutions, etc.
4	GPE	Countries, cities, states
5	EVENT	Named hurricanes, battles, sports event, etc.
6	PRODUCT	Objects, vehicles, foods, etc.

As described in Table II, experiments are conducted on four datasets in which each document only belongs to one category. The first dataset (DS1), second dataset (DS2), third dataset (DS3), and fourth dataset (DS4) contains 200, 500, 1000 and 5000 random documents belonging to 4, 5, 8, and 15 categories respectively. The distribution of categories in DS4 is highly skewed. As mentioned above, six specific types of named entities are counted and the total number of them contained in each dataset is also listed in Table II.

TABLE II. THE DETAILS OF THE DATASETS

Datasets	**# of documents**	**# of named entities**	**# of clusters**
DS1	200	1514	4
DS2	500	3878	5
DS3	1000	9578	8
DS4	5000	42223	15

In our experiments, five classical evaluation criteria are adopted to measure the performance of clustering, including Accuracy, Precision, Recall, F1 score and Adjusted Rand Index (ARI).

B. Results and Disscusion

As shown in Table III, the experimental results demonstrate that methods with WA and WR schemes which are proposed in this paper both achieve higher accuracy than the unweighted average on all four datasets. The WA scheme is simple but comparatively effective, especially out-performs the other two schemes on smaller datasets containing hundreds of documents. The WR scheme associates each document with the entire document collection, which has better performance on larger datasets.

The statistical analysis (Friedman test) is carried out using the values of F1 score and ARI separately. The minimum ranking value indicates the best scheme. The average rankings of methods with three schemes are listed in Table IV and Table V.

TABLE III. PERFORMANCE COMPARISONS

Method		**UA**	**WA**	**WR**
DS1	*Accuracy*	0.7500	**0.7550**	**0.7550**
	Precision	0.7668	**0.7757**	0.7717
	Recall	0.7500	**0.7550**	**0.7550**
	F1	0.7453	**0.7492**	0.7490
	ARI	0.5101	0.5142	**0.5197**
DS2	*Accuracy*	0.6200	**0.6300**	0.6220
	Precision	**0.5464**	0.5418	0.5433
	Recall	0.6200	**0.6300**	0.6220
	F1	0.5618	**0.5669**	0.5620
	ARI	0.3791	**0.4050**	0.3829
DS3	*Accuracy*	0.5320	0.5330	**0.5370**
	Precision	0.4322	0.4270	**0.4469**
	Recall	0.5320	0.5330	**0.5370**
	F1	0.4488	0.4474	**0.4537**
	ARI	0.3166	**0.3267**	0.3172
DS4	*Accuracy*	0.6772	**0.6792**	0.6776
	Precision	**0.2397**	0.2112	0.2394
	Recall	0.3194	0.3166	**0.3196**
	F1	**0.2535**	0.2435	**0.2535**
	ARI	0.7020	0.7002	**0.7036**

TABLE IV. THE AVERAGE RANKING BASED ON F1 SCORE

Method	**DS1**	**DS2**	**DS3**	**DS4**	**Mean Rank**	**Ranking**
UA	3	3	2	1	2.25	3
WA	1	1	3	3	2	2
WR	2	2	1	1	1.75	1

TABLE V. THE AVERAGE RANKING BASED ON ARI

Method	**DS1**	**DS2**	**DS3**	**DS4**	**Mean Rank**	**Ranking**
UA	3	3	3	2	2.75	3
WA	2	1	1	3	1.75	2
WR	1	2	2	1	1.5	1

As indicated in two tables above, the proposed WR scheme is ranked the highest among the four datasets, which is followed by WA scheme, and UA method without using any weighting scheme. Our proposed method with WR scheme gets better overall performance among all datasets.

IV. CONCLUSIONS

In this paper, we present a novel text document clustering method. Firstly, pre-trained language representation model BERT is utilized to generate contextualized sentence embeddings. Then, two sentence-level weighting schemes namely WA and WR are designed based on named entity in order to enhance the performance. Finally, the k-means clustering algorithm is applied to find groups of similar documents. Experimental results on four datasets demonstrate that our weighted method obtains higher accuracy than unweighted average method. Friedman tests conducted separately with F1 score and Adjusted Rand Index (ARI) values validate better overall performance of our proposed method. For future work, we will do research on sentence features and involve them to weighting schemes in our method to improve its overall performance.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (Grant No. 61631016, 61971383) and Horizontal Project (HG19006).

* Jingling Wang is the corresponding author

References

[1] M. Afzali and S. Kumar, "Text Document Clustering: Issues and Challenges," 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 2019, pp. 263-268.

[2] M. Sukanya and S. Biruntha, "Techniques on text mining," 2012 IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), Ramanathapuram, 2012, pp. 269-271.

[3] N.A. Smith, "Contextual Word Representations: A Contextual Introduction," arXiv preprint arXiv:1902.06006, 2019.

[4] Q. Le and T. Mikolov, "Distributed Represenationsof Sentences and Documents, " In Proceedings of ICML, 2014.

[5] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," Advances in neural information processing systems, 2013, pp. 3111-3119.

[6] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532-1543.

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

[8] A. Vaswani, N. Shazeer, N. Parmar, et al. "Attention is all you need," Advances in neural information processing systems, 2017.

[9] Y. Wu, M. Schuster, Z. Chen, et al. "Google's neural machine translation system: Bridging the gap between human and machine translation," arXiv preprint arXiv:1609.08144, 2016.

[10] D. Jurafsky and J.H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition, Second Edition. Upper Saddle River, NJ: Prentice Hall, 2009.

[11] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," International Conference on Learning Representations, 2017.

[12] L.M. Abualigah and A.T. Khader, "Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering," The Journal of Supercomputing 73.11, 2017,pp. 4773-4795.