American Journal of Information Science and Technology
2019; 3(3): 62-71
http://www.sciencepublishinggroup.com/j/ajist
doi: 10.11648/j.ajist.20190303.12
ISSN: 2640-057X (Print); ISSN: 2640-0588 (Online)

MLPV: Text Representation of Scientific Papers Based on Structural Information and Doc2vec

Yonghe Lu*, Yuanyuan Zhai, Jiayi Luo, Yongshan Chen

School of Information Management, Sun Yat-sen University, Guangzhou, China

Email address:
luyonghe@mail.sysu.edu.cn (Yonghe Lu), zhaiyy5@mail2.sysu.edu.cn (Yuanyuan Zhai)

*Corresponding author

To cite this article:
Yonghe Lu, Yuanyuan Zhai, Jiayi Luo, Yongshan Chen. MLPV: Text Representation of Scientific Papers Based on Structural Information and Doc2vec. *American Journal of Information Science and Technology*. Vol. 3, No. 3, 2019, pp. 62-71. doi: 10.11648/j.ajist.20190303.12

Received: July 15, 2019; **Accepted**: August 12, 2019; **Published**: August 28, 2019

Abstract: Text representation is the key for text processing. Scientific papers have significant structural features. The different internal components, mainly including titles, abstracts, keywords, main texts, etc., embody different degrees of importance. In addition, the external structural features of scientific papers, such as topics and authors, also have certain value for analysis of scientific papers. However, most of the traditional analysis methods of scientific papers are based on the analysis of keyword co-occurrence and citation links, which only consider partial information. There is a lack of research on the textual information and external structural information of scientific papers, which has led to the inability to deeply explore the inherent laws of scientific papers. Therefore, this paper proposes Multi-Layers Paragraph Vector (MLPV), a text representing method for scientific papers based on Doc2vec and structural information of scientific papers including both internal and external structures, and constructs five text representation models: PV-NO, PV-TOP, PV-TAKM, MLPV and MLPV-PSO. The results show that the effect of the MLPV model is much better than the PV-NO, PV-TOP and PV-TAKM models. The average accuracy of MLPV model is much more stable and higher, reaching 91.71%, which proves its validity. On the basis of the MLPV model, the accuracy of the optimized MLPV-PSO model is 3.33% higher than MLPV model which proves the effectiveness of the optimization algorithm.

Keywords: MLPV Model, Scientific Papers, Text Representation, Doc2vec, Structural Features

1. Introduction

Before performing natural language processing (NLP), as an unstructured data, text needs to be transformed into structured data that can be recognized by computers, which is called text representation. Text representation is the basic and import part of NLP. The quality of text representation has directly influence on the effectiveness of text semantic analysis, such as text classification, text clustering, automatic extraction of summary and keywords, and calculation of text similarity. Therefore, it has caught extensive attention of scholars and has made great progress. The traditional text representation models which have been widely used mainly include Boolean logic model, probability model, vector space model and N-gram model. Recent research is mostly based on the distributed representations of individual words or continuous words, or based on deep learning.

A scientific paper is a special kind of text, which has a fixed drafting standard. With the development of science and technology, the number of scientific papers has been increasing dramatically. However, most of the traditional methods for analyzing scientific papers are mainly based on keyword co-occurrence and link information, which only consider partial information [1-3]. Ming Liu et al. take partial structural information such as target, methodology, domain, style, date and keywords into consideration, which is called semantic Profile [4]. Mahdi A E extracted and constructed a set of key phrases from the references, so as to mark the key words and automatically index documents [5]. There is a lack of research on the textual information and external structural information of scientific papers, which has led to the inability to deeply explore the inherent laws of scientific papers. Nowadays, the development of data mining and natural language processing technology has greatly enriched the theory of text processing and text representation. Using these

new techniques to analysis scientific papers can, to some extent, make up for the deficiencies of traditional analytical methods. This paper proposes an improved text representation method based on Doc2vec and structural information for scientific papers. Different from the traditional methods which consider text as a whole and lack consideration of structural information, this method not only retains the semantic information, but also considers the structural features of scientific papers.

2. Related Work

The vector space model takes every individual feature word as an individual feature item in the vector space. It assumes that feature words are independent of each other. This model is characterized by its simplicity, but it ignores the semantic information between feature words and often leads to spatial dimensional catastrophes, causing difficulties for further processing. Faced with this deficiency, some scholars proposed topic models, mainly including Latent Semantic Indexing (LSI), Probabilistic Latent Semantic Indexing (PLSI), and Latent Dirichlet Allocation (LDA) [6-9]. Luo L et al. present a method which combines the Latent Dirichlet Allocation (LDA) algorithm and the Support Vector Machine (SVM) to achieve high performance in classification process [10]. However, the calculation of the topic model takes a lot of time. Since topic model takes phrases as feature items, it may not be able to capture enough semantic information.

Word vector is a very popular method for text representation, of which One-hot Representation is the simplest form. In this formal expression, the dimensionality of the word vector is the size of the vocabulary. Although the One-hot Representation is simple, it has some limitations: 1) it is easy to cause dimensional disaster problems because the vocabulary of the text set is very large; 2) there is a "vocabulary gap". This method assumes that any two words are independent, which is inconsistent with the actual situation and cannot reveal the similarity and relevance among words.

To make up for the deficiencies of the One-hot Representation, Hinton proposed the concept of Distributed Representation in 1986 [11]. The basic idea of this method is to map each word into a K-dimensional real number vector, where each dimension of the vector represents abstract semantic information. There are two kinds of models for training the word vectors of the Distributed Representation: the word vector training model based on the matrix decomposition and the word vector training model based on the neural network. The former mainly includes latent semantic models. The latter mainly includes the neural network language model (NNLM) proposed by Bengio et al. [12]. After that, Google proposed word2vec [13, 14]. Word2vec includes continuous bag-of-words models (CBOW) and continuous skip-gram models (skip-gram). Word2vec can learn high-quality word vectors from a large-scale text set in a short period of time, and can describe the semantic information of words very well. Therefore, it has been applied in various aspects of text processing tasks such as word clustering, synonym expansion, topic recommendation and text representation [15-17]. To better understand text from the perspective of semantic logic, some researches have been done by means of external knowledge base and word embedding [18-20].

Although the word2vec model averages word vectors, it still ignores the influence of word order. Mikolov et al. proposed Doc2vec in 2014, inspired by word2vec [21]. In addition to adding a paragraph vector, Doc2vec is almost equivalent to word2vec. Compared to word2vec, Doc2vec has contextual "semantic analysis" capabilities.

Doc2vec algorithm can obtain a fixed-length vector representation of a document from a large-scale text set in a short time, and can describe the semantic information of paragraphs or texts. More and more researches are based on Doc2vec. In the study of algorithm availability, Dai et al. compared the effects of LDA, word bag model, mean word vector, and paragraph vector on two semantic analysis tasks, pointing out the paragraph vector is superior to other models, with an accuracy of 93% [22]. Gabriele Fisher et al. compared the paragraph vector, the LDA model, and the traditional word2vec model in the Wikipedia navigation experiment, and established three extensions to the original paragraph vector model [23]. It was found that combinations of paragraph structures assisted in optimizing Paragraph Vector training. In the improvement of the algorithm, Grzegorczyk et al. proposed a binary paragraph vector, and introduced a Sigmoid nonlinear method to extend the paragraph vector [24]. Experiments have proved that this method was much better than the self-encoded binary code segment. Palangietal found that applying bidirectional-LSTM-RNNs to the paragraph vector in information retrieval could improve the accuracy rate by 5.2% [25].

Scientific papers have significant structural features. The different internal components, mainly including titles, abstracts, keywords, main texts, etc., embody different degrees of importance. However, the traditional Doc2vec model directly trains the entire text as a whole, ignoring the differences between different text blocks. In addition, the external features of scientific papers, such as topics and authors, also have certain research value for text analysis of scientific papers. In the process of learning, adding external structural information will make text analysis results more scientific. This paper proposes Multi-Layers Paragraph Vector (MLPV) model based on structural information to represent scientific papers and the Paragraph Vector model (PV model, Doc2vec). Different from the PV model, the MLPV model divides the text into different text blocks according to the internal structures and trains them separately and the external structure identifier of the text block is added during the training. According to certain rules, the vectors are generated by concatenating the vectors of different text blocks.

3. PV Model

PV model, also called Doc2vec, is an unsupervised framework that can learn fixed-length feature representations

from variable-length pieces of text, such as sentences, paragraphs, chapters and documents. Similar to word2vec, PV model has two models as well, Distributed Memory (PV-DM) and Distributed Bag of Words (PV-DBOW).

Figure 1 shows the two models of PV model [21]. The top layer of the model is the input layer, the middle layer is the hidden layer, and the bottom layer is the output layer. The PV-DM model predicts the next word when given the average or concatenation of the paragraph vector and word vectors in a context, while the PV-DBOW model predicts a set of random words in a paragraph when given only the paragraph vector.

Take PV-DM as an example. First, every paragraph is mapped to a unique vector, represented by a column in matrix D and every word is mapped to a unique vector, represented by a column in matrix W. Second, the paragraph vector and word vectors are averaged or concatenated as an input to the softmax layer to predict the next word in a context.

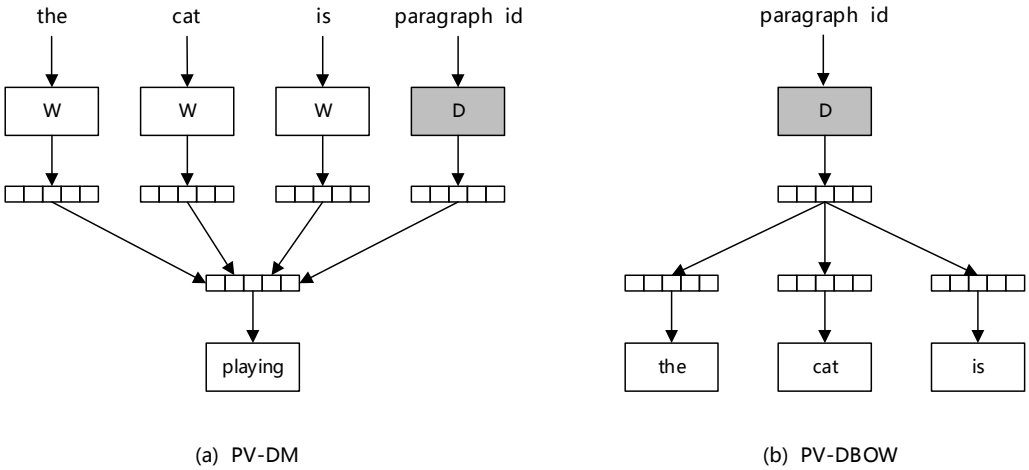

Figure 1. *The PV-DM model (left) and the PV-DBOW model (right).*

4. MLPV Model and Its Optimization

The information of different structures within scientific papers has different importance. If the PV model is used directly, scientific paper is treated as a whole during the training process, which will ignore the structured information. Therefore, based on the traditional PV model, this paper constructs the MLPV model and explores its optimization to better represent scientific papers.

4.1. MLPV Model

MLPV model integrates the structural information of scientific papers into the PV model. The specific algorithm flow is as follows:

Step 1: Get internal structure information. Scientific papers generally exist in the form of PDF. By parsing the PDF files, the paper is divided into a plurality of text blocks according to the font size and the HTML identifier.

Step 2: Get external structure information. We can write a web crawler or use crawler software to get external structure information, such as authors, publishers, etc.; for some information that cannot be obtained directly, such as topic information, we can conduct machine learning methods (LDA, LSI, etc.) to get them.

Step 3: Train text blocks separately. Use the PV model to train each text block obtained in step 1, and set the paragraph id as the identifier of the external structure and obtain the paragraph vector v_i of the text block i.

Step 4: Concatenate text block vectors. Concatenate the text block vectors obtained in step 3 according to certain rules. The most primitive rule is directly concatenating the vectors in the order of the text blocks to obtain the final text vector representation (v_1, v_2,..., v_i,..., v_n).

The diagram of the MLPV model is shown in Figure 2.

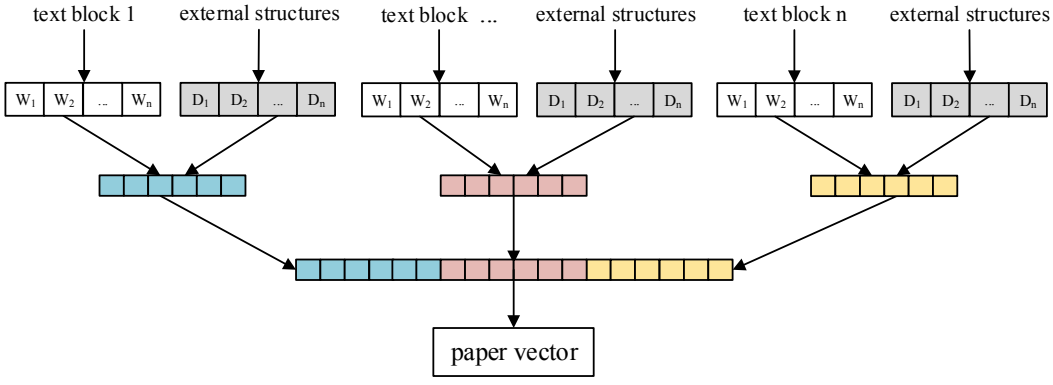

Figure 2. *MLPV model.*

4.2. Optimization of MLPV Model

The last step of the MLPV algorithm requires concatenating the vectors of different text blocks in accordance with certain rules. The simplest way is directly concatenating, but there is certain different importance of different blocks in the representation of scientific papers. Therefore, when concatenating the structural vectors, we try to introduce the position weight coefficient to adjust the ratio of the vectors, and use the particle swarm algorithm (PSO) to optimize the coefficient combination for seeking a better concatenating method [26]. Assuming the paper is divided into four text blocks, the specific algorithm flow for model optimization is as follows:

Step 1: According to the MLPV model, get the vectors of the paper (v_1, v_2, v_3, v_4).

Step 2: Add the position weight coefficients a, b, c, d, and concatenate the vectors of text blocks according to the following rules: $MLPV\text{-}PSO = (a^* v_1, b^* v_2, c^* v_3, d^* v_4)$.

Step 3: Use the PSO algorithm to find the optimal position weight coefficient combination a, b, c, d, denoted as (*abest, bbest, cbest, dbest*).

Step 4: According to the methods in step 2 and 3, the document vector of scientific paper will be represented as: $docVecPSO = (abest^*v_1, bbest^*v_2, cbest^*v_3, dbest^*v_4)$.

5. Experiments

In order to verify the validity of the MLPV model, this section will conduct experiments based on real data.

The experiments in this paper run on Windows 10_64 bit system, the hardware configuration is i5 processor, 8GB memory. All experiments were programmed in Python and the IDE was PyCharm. The word segmentation used Python-based jieba module, and Gensim's Doc2vec toolkit was used and run on a Linux virtual machine when running the Doc2vec model.

5.1. Data Collection and Preprocessing

For building datasets, all papers in 10 core journals in the field of Information Science in Chinese Social Science Citation Index (CSSCI) from 2011 to 2016 are crawled, including "Information Science", "Information studies: Theory & Application", "Journal of The China Society for Scientific and Technical Information", "Journal of Intelligence", " Information and Documentation Services", "Library and Information Service", "Document, Information & Knowledge Library", "Library and Information", " Modern Information" and "New Technology of Library and Information Service". When crawling papers, we downloaded PDF files of the papers and saved the journal name, year and period number. The total number of papers crawled initially was 18,075.

For general texts, the preprocessing procedure mainly focuses on text segmentation and removal of stop words. However, for scientific papers, because their structural features are different from general texts, we proposes a method for preprocessing large-scale scientific papers, as shown in “Figure 3”.

Firstly, use the PDF2Text tool to convert the PDF format into TXT format. Since the font sizes of different text blocks in TXT format are different, the structure of scientific papers can be identified automatically, and data cleaning is performed to remove papers with unrecognizable structures. Afterwards, the keywords of all scientific papers are extracted to form a keyword set, which is imported into the jieba text segmentation module as a user dictionary later. Finally, segment the texts, remove stop words and filter speech pattern. After the data processing, there were 16,376 articles stored in the database.

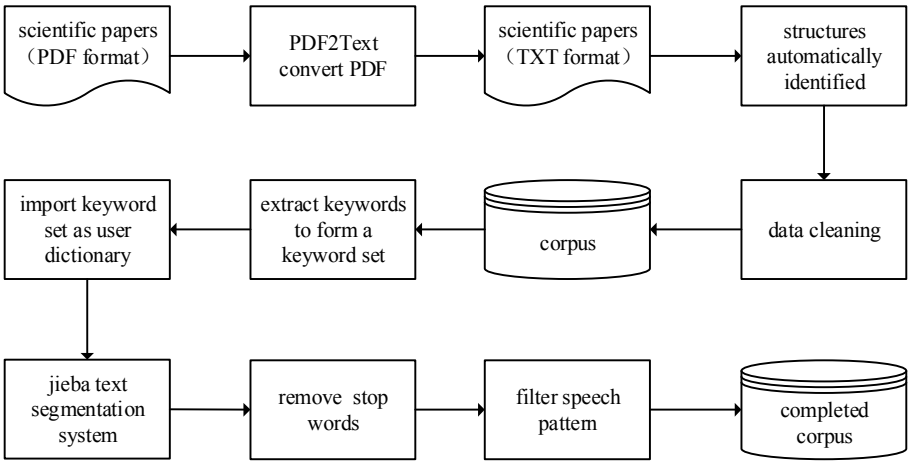

Figure 3. *The flow of preprocessing large-scale scientific papers.*

5.2. Structure Information Extraction

After the preprocessing, internal structures of scientific papers have been automatically identified. This section mainly introduces how to obtain external structure information. The external structure of this experiment is the topic of the paper obtained by the combination of LDA model and manual annotation.

5.2.1. LDA Model

Clustering: Use LDA model to cluster scientific papers. The clustering iteration was set to 5,000. In each experiment, the number of clusters was set to an integer in 9~25, and a total of 17 experiments were performed. However, how many clusters are reasonable? It will be determined by the effect of self-classification.

Evaluation based on self-classification: After clustering, every text already has a cluster label. First, with different random seeds, divide them into two parts. 60% of the texts were taken as training set and 40% were taken as testing set. Second, used Chi-square for feature selection, set the feature dimension to 10000~27500 (each span is 2500), and then used the TFIDF model for text representation. Finally, Naïve Bayes (NB) classifiers and Support Vector Machine (SVM) classifiers were used for classification. The accuracy of self-classification was used as a criterion to evaluate the effect of LDA clustering. The average accuracy of self-classification under different cluster numbers is shown in Figure 4.

As the number of clusters increases, the average accuracy of both classifiers tends to decrease (Figure 4). However, when the number of clusters is 20 and 23, their accuracies obviously increase. This indicates that when the number of clusters is 20 or 23, LDA model perform better. Especially, the accuracy is higher when the number of clusters is 20, with an average of 87.34% in SVM and 83.65% in NB. Therefore, the clustering result with 20 clusters is finally selected as the reference for the manual annotation.

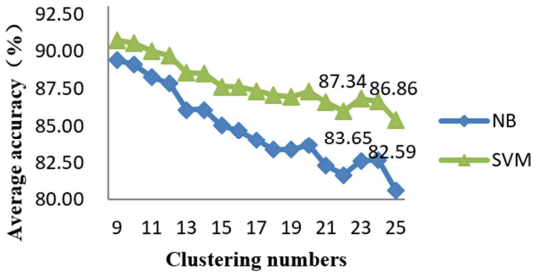

Figure 4. *The average accuracy of self-classification under different clustering numbers.*

5.2.2. Manual Annotation

Table 1 shows the No., topic and number of texts for each cluster.

Table 1. *Topic information of LDA clustering.*

No.	Topic	Total	No.	Topic	Total
1	Evaluation index system construction	803	11	Information Communication and Public Opinion Analysis	1069
2	Information literacy education and personnel training	614	12	Social Media and User Research	732
3	Knowledge ontology	886	13	E-government and social information construction	522
4	Patent analysis	684	14	Information legal system construction	914
5	Competitive Intelligence Analysis	681	15	Information Resource Management and Sharing	999
6	E-commerce and Information Industry Research	512	16	Knowledge Management and Knowledge Sharing	1257
7	Data mining	1280	17	Information Retrieval and Information System Design	461
8	Information Ecology Chain and Information Science Theory	615	18	Social Network and Complex Network	1085
9	Library Digital Resources and Discipline Service Construction	1325	19	Library Collections and Reading Promotion	1688
10	Ancient Books and Historical Materials	274	20	Bibliometric analysis	1451

Observing the topic information of 20 clusters, it can be found that the texts of Category 1 can be divided into other categories, and there are also some misclassified texts in all categories. In order to obtain more precise topic labels, two labelers manually classified all scientific papers into the categories of their corresponding topics according to their titles, abstracts, keywords and the topic words of categories. Result of manual annotation is shown in the Table 2 (see the Appendix for details). Cohen's kappa coefficient is used to estimate whether the two labelers' outputs are consistent. The kappa coefficient of the manual annotation is 0.857, which means that the quality of the manual annotation is excellent and meets the requirements.

Table 2. *Manual annotation result.*

B labeler \ A labeler	1	2	3	4	5	6	7	8	9	10
1	555			3			6	28	2	
2	1	750		2	2	21	8	9		2
3		4	663	10	5	16		6		
4	1	2	26	559	41	7	4	14	1	3
5		1	9	18	451	1	2	8	1	1
6	1	11	3			1193	4			6
7							519			
8	11	8	11		3	4	2	942	3	
9	1						5	7	251	
10						5	5		1	738

Randomly select the results of manual annotation of two labelers for further observation. It was found that the labeling

of B labeler was more reasonable and reliable, so the result of the B labeler was selected as the final labeling result. The labeling result after LDA clustering and manual labeling (B labeler) is shown in Table 3.

Table 3. *Topic information after LDA clustering and manual labeling.*

No.	Topic	Total	No.	Topic	Total
1	Information literacy education and personnel training	645	11	Social Media and User Research	1069
2	Knowledge ontology	837	12	E-government and social information construction	732
3	Patent analysis	731	13	Information legal system construction	522
4	Competitive Intelligence Analysis	745	14	Information Resource Management and Sharing	914
5	E-commerce and Information Industry Research	572	15	Knowledge Management and Knowledge Sharing	999
6	Data mining	1268	16	Information Retrieval and Information System Design	1257
7	Information Ecology Chain and Information Science Theory	519	17	Social Network and Complex Network	461
8	Library Digital Resources and Discipline Service Construction	1041	18	Library Collections and Reading Promotion	1085
9	Ancient Books and Historical Materials	272	19	Bibliometric analysis	1688
10	Information Communication and Public Opinion Analysis	767	N	Unclassifiable	252

Note: N indicates the situation where it is difficult to judge which category the paper should belong to.

5.3. Models and Parameter Settings

5.3.1. Models

Six models are involved in this experiment, including the traditional TF-IDF model, the original PV model (hereafter referred to as PV-NO), and four proposed models.

TF-IDF: Use the TF-IDF model for text representation. And it treats scientific papers as a whole without considering any internal and external structural information.

PV-NO: Use the Doc2vec model for text representation. And it also treats scientific papers as a whole without considering any internal and external structural information

PV-TOP: Add the text topic identifier when training PV-NO model. It treats scientific papers as a whole as well, but it considers the topics of scientific papers (the external structure).

PV-TAKM: Use the Doc2vec model to represent the content of 4 parts of the text respectively, including the title, abstract, keywords, and main text, and directly concatenate these four vectors as the document vector. This model has considered the internal structural information of scientific papers.

MLPV: Add the text topic identifier when training PV-TAKM. This model considers both external and internal structural information on scientific papers.

MLPV-PSO: Introduce position weight coefficients when concatenating vectors, and PSO is used to optimize the coefficients. This model not only considers the external and internal structural information of scientific papers, but also considers that different internal structures contribute differently to scientific papers.

5.3.2. Parameter Settings

MLPV model used in this experiment includes four internal text blocks: title, abstract, keyword, and main text. Its external structural feature is the topic identifier. The model is shown in Figure 5.

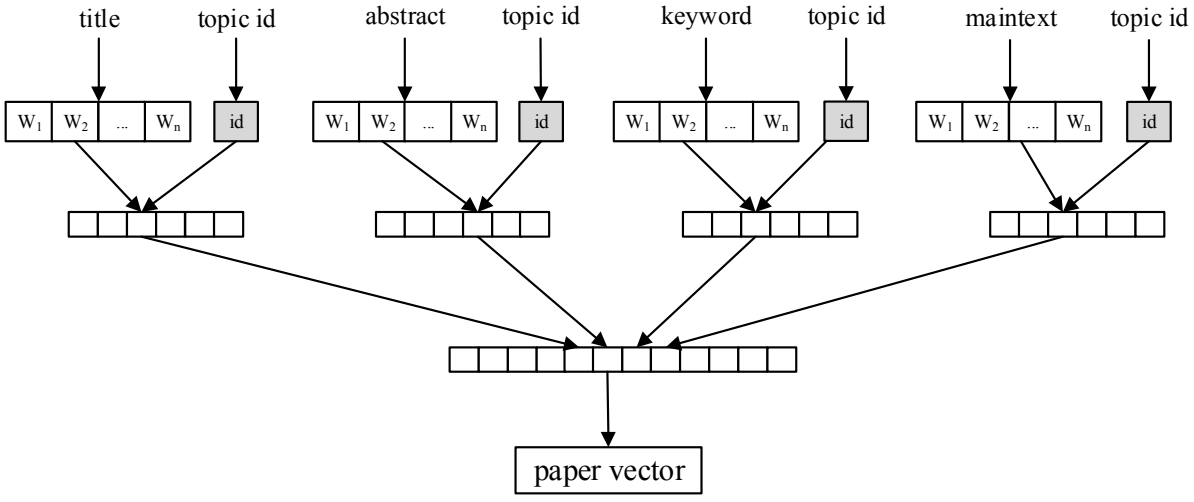

Figure 5. *MLPV model for experiment.*

Through a lot of pre-experiments, the parameter settings of the MLPV model are shown in Table 4. The parameter settings of the PV, PV-TOP, PV-TAKM, and MLPV-PSO models are consistent with the MLPV model. Besides, when training MLPV-PSO, the number of iterations of PSO is set to 200.

Table 4. *Topic information of LDA clustering.*

Parameter	Parameter Setting
-window	10
-iter	10
-size	100
-min_count	5
-dm	0
-dm_concat	1
-hs	0
-negative	5
-sample	1e-3
- workers	16

5.4. Evaluation Method

The experimental task of this paper is text semantic similarity measurement. According to the evaluation method of the paper [22], based on the topic clustering, we try to construct a triple set. Three papers form a triple. The first two texts in each triple are similar while the third text is not similar to the first two texts. Then we use the evaluation function of the triple as the evaluation method of the text representation.

On the basis of topic clustering, the first two papers in the triple are randomly selected from the same category, and the third is randomly selected from the rest categories. A total of 3700 triples were generated to evaluate the effects of different models in section 5.3.1. The effect of the text representation method is calculated by the evaluation function of the triple set, which is defined as:

$$\text{Accuracy}\ () = \frac{\text{the number of triples where } \cos(\text{p1,p2}) > cos(\text{p1,p3})}{\text{total number of triples}} \quad (1)$$

Among them, cos (p1, p2) represents the cosine distance between the first paper and the second paper, and cos (p1, p3) represents the cosine distance between the first paper and the third paper.

5.5. Experiment

TF-IDF model was used as baseline with the feature dimension ranging from 100 to 2500 (each span is 100). In order to verify the MLPV model, the following two experiments were performed on the same triple set:

5.5.1. Experiment A: Fixed Epochs, Variable Size

Table 5. *Specific sub-structure size settings.*

Size	Title	Abstract	Keyword	Main Text
100	10	20	10	60
200	20	40	20	120
300	30	60	30	180
400	40	80	40	240
500	50	100	50	300

Based on a large number of pre-experiments, it is found that when the number of iterations is 10, the result is better. In this experiment, the epoch is fixed at 10, and the vector size is set to 100, 200, 300, 400, and 500, respectively. Considering the size of the text blocks, we set the ratio of the dimensions of the title, abstract, keyword and main text to 1:2:1:6. The details are shown in Table 5. Finally, each paper is represented by the combination of 4 vectors: the vectors of its title, abstract, keyword and main text.

5.5.2. Experiment B: Fixed Size, Variable Epochs

Based on a large number of pre-experiments, it is found that when the number of vector size is 100, the result is better. In this experiment, the vector size is fixed at 100, and the number of epoch is set from 1 to 20 (each span is 1). Considering the size of the text blocks, we set the vector sizes in the ratio of 1:2:1:6, that is, set the size of the title vector to 10, the size of the abstract vector to 20, and the size of the keyword vector to 10 and the size of the main text vector to 60. Finally, each paper is represented by the combination of 4 vectors: the vectors of its title, abstract, keyword and main text.

6. Result and Analysis

6.1. Baseline: TF-IDF Model

As shown in Figure 6, the accuracy of the TF-IDF model fluctuates with the change of the feature dimensions, and the data shows a trend of rising first and then stable fluctuation on the whole. One of the reasons may be that when the number of features is small, the performance of TF-IDF model becomes better with the increase of effective features. When the features reach a certain level, redundant features will be introduced as the number of features increases, which will worsen the effect of TF-IDF model. When the feature dimension is 800, the accuracy rate reaches a maximum of 58.35%. Overall, the accuracy rate of TF-IDF model fluctuates from 56% to 59%.

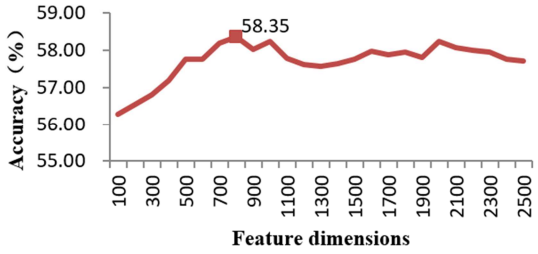

Figure 6. *The accuracy rate of TF-IDF model.*

6.2. Experiment A: Fixed Epochs, Variable Size

In Experiment A, as shown in Figure 7, as the size of the document vector increases, the accuracies of the PV-NO, PV-TOP, PV-TAKM, MLPV, and MLPV-PSO models are much greater than TF-IDF model. That is, these five models work better than TF-IDF model in text representation. PV-TAKM model is better than the original PV-NO model, verifying the effect of dividing scientific papers according to their internal structures. The effect of PV-TOP model is worse than PV-NO model. One of the possible reasons is that the number of epoch is not enough and the algorithm has not converged yet. Obviously, the effectiveness of the MLPV model is much better than the above three models, with an

average accuracy of 91%. On this basis, the accuracy of the MLPV-PSO model is improved by an average of 4.2% compared with the MLPV model, which proves that the introduction of the position weight coefficient assists in optimizing MLPV algorithm.

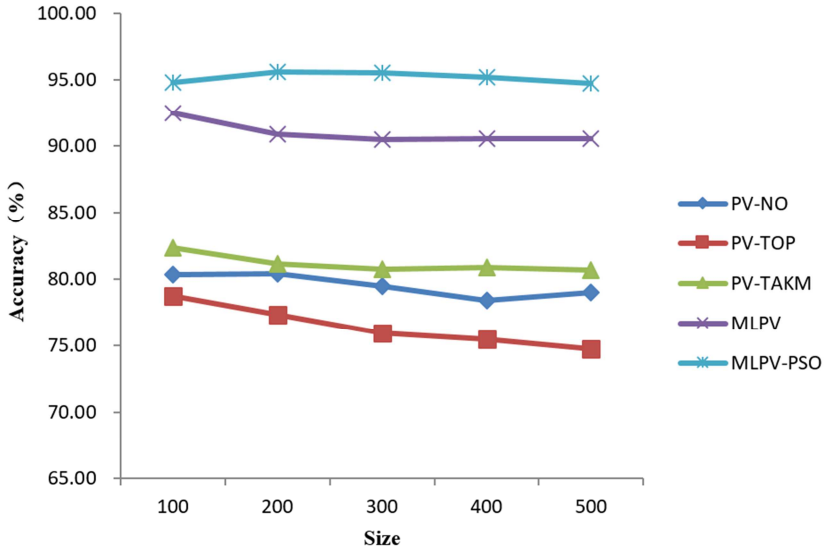

Figure 7. *The result of Experiment A.*

6.3. Experiment B: Fixed Size, Variable Epochs

In Experiment B, as shown in Figure 8, the accuracies of the PV-NO, PV-TOP, PV-TAKM, MLPV, and MLPV-PSO models are also much greater than the accuracy of TF-IDF model, which means that the five models are again proven to be more effective than TF-IDF in the field of text representation of scientific papers.

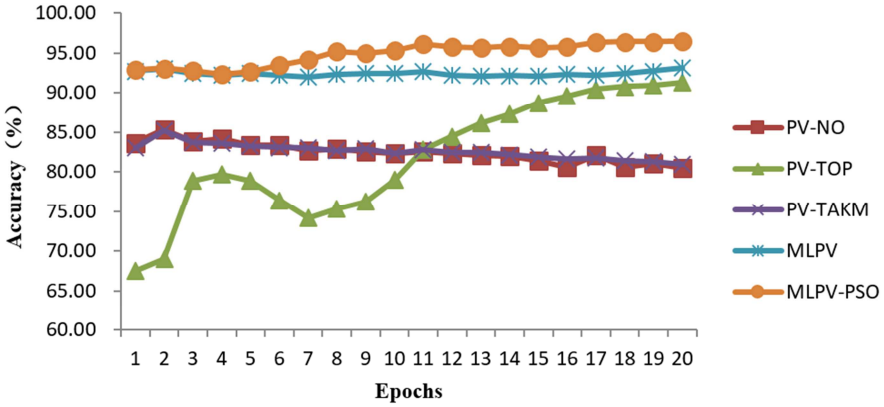

Figure 8. *The result of Experiment B.*

The effectiveness of the PV-TAKM model is almost the same as PV-NO model, and is stable to some extent. The accuracy of PV-TOP model fluctuates greatly. When the number of epoch is less than 10, the accuracy of PV-TOP model is lower than PV-NO model. When the number of epoch is greater than 10, the accuracy of the model rises rapidly and is higher than PV-NO. That shows the validity of introduction of topic identifier, and also explains why the effect of PV-TOP model in Experiment A is not as good as PV-NO model. Similarly, MLPV model is much better than the above three models, with an average accuracy of 92.42%. What's more, the accuracy of MLPV-PSO model is 2.46% higher than MLPV model. It also proves that the introduction of the position weight coefficient is effective for optimizing MLPV algorithm.

7. Conclusion

Different from general texts, scientific papers are strongly structured. This paper proposes Multi-Layers Paragraph Vector (MLPV), an improved text representation model based on text structure information and Doc2vec, for text representation of scientific papers. PV-NO, PV-TOP, PV-TAKM, MLPV, and MLPV-PSO were constructed and two sets of comparative experiments were designed to verify the validity of the model. The results show that the effectiveness of MLPV model is much better than PV-NO, PV-TOP, PV-TAKM model with an average accuracy of 91.71%. And the accuracy of MLPV-PSO model is 3.33% (from the average of 4.2% and 2.46%) higher than MLPV

model, which proves that the introduction of position weight coefficient assists in optimizing MLPV algorithm. In future research, the main text block of papers can be subdivided, and more external structural information can be introduced to text representation of scientific papers.

Acknowledgements

This research was supported by Science and Technology Planning Project of Guangdong Province, China (Grant No. 2016B030303003).

Appendix: Manual Labeling Result

The following table shows the results of the two labelers. Due to the limitation of the space of the text, the original data is split into two parts, and the data of the two parts together constitute the manual annotation result, as shown in Table 6 and Table 7.

Table 6. *Manual annotation result.*

B labeler \ A labeler	1	2	3	4	5	6	7	8	9	10
1	555			3			6	28	2	
2	1	750		2	2	21	8	9		2
3		4	663	10	5	16		6		
4	1	2	26	559	41	7	4	14	1	3
5		1	9	18	451	1	2	8	1	1
6	1	11	3			1193	4			6
7							519			
8	11	8	11		3	4	2	942	3	
9	1						5	7	251	
10						5	5		1	738
11	1	4				1	11	18		39
12	1	1	1	12	3	2	5	10	3	12
13			2	2		1	1	1		
14	2	15		10	5		11	159	4	
15		17	1	5	6	1	10	52		
16	3	24		17	2	36	5	24	3	1
17		4		2	1	4		7		5
18	4	1		2			2	55	4	
19		2	6	8		2	1	5	1	2
N	3	4	3	7	4	1	14	37	3	11
Sum	584	848	725	657	523	1295	615	1382	277	820

Table 7. *Manual annotation result.*

B labeler \ A labeler	11	12	13	14	15	16	17	18	19	N	Sum
1	5	4	4	20	2	3		4	6	3	645
2	4		2	6	4	12	3	3	6	2	837
3		4			2	7	8		6		731
4	7	10	2	8	17	4	3		14	22	745
5	23	31	7	2	9			3	1	4	572
6	7	4	1	4		19	4	2	8	1	1268
7											519
8	7	1	1	10	4	20		10	4		1041
9		1						6	1		272
10	8	2			1	2	3		1	1	767
11	938	13	2	10	7	9	5	9	2		1069
12	16	597	24	29	3	5		6	1	1	732
13		7	473	21	1	1		9	1	2	522
14	5	45	14	579	19	27	5	6	7	1	914
15	20	3		9	839	7	13		14	2	999
16	10	4	2	5	8	1097	4	4	6	2	1257
17	1	1		1	11	2	414		8		461
18	11	22		9	3	7	1	961	3		1085
19	4	5	2	6	8	4	14	3	1612	3	1688
N	9	26	4	16	18			31	16	45	252
Sum	1075	780	538	735	956	1226	477	1057	1717	89	16376

References

[1] Yoon S H, Kim J S, Kim S W and Lee C. TL-Rank: A Blend of Text and Link Information for Measuring Similarity in Scientific Literature Databases [J]. IEICE TRANSACTIONS on Information and Systems, 2012, 95 (10): 2556-2559.

[2] Hamedani M R, Kim S W, Kim D J. SimCC: A novel method to consider both content and citations for computing similarity of scientific papers [J]. Information Sciences, 2016, 334: 273-292.

[3] Cao M, Sun X, Zhuge H. The contribution of cause-effect link to representing the core of scientific paper—The role of Semantic Link Network [J]. PloS one, 2018, 13 (6): e0199303.

[4] Liu M, Lang B, Gu Z and Zeeshan A. Measuring similarity of academic articles with semantic profile and joint word embedding [J]. Tsinghua Science and Technology, 2017, 22 (6): 619-632.

[5] Mahdi A E, Joorabchi A. A citation-based approach to automatic topical indexing of scientific literature [J]. Journal of Information Science, 2010, 36 (6): 798-811.

[6] Xu G, Wang H F. Development of topic models in natural language processing. [J]. Chinese J Comput, 2011 (8): 1423-1436. M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 198.

[7] Deerwester S, Dumais S T, Furnas G W, Landauer T K, Harshman R. Indexing by latent semantic analysis [J]. Journal of the Association for Information Science & Technology, 1990, 41 (6): 391-407.

[8] Hofmann T. Probabilistic latent semantic indexing [C]// International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 1999: 50-57.

[9] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation [J]. J Machine Learning Research Archive, 2003, 3: 993-1022.

[10] Luo L, Li L. Defining and evaluating classification algorithm for high-dimensional data based on latent topics [J]. PloS one, 2014, 9 (1): e82119.

[11] Hinton G E. Learning distributed representations of concepts. [C]// Eighth Conference of the Cognitive Science Society. 1986.

[12] Bengio Y, Ducharme R, Vincent P, Jauvin C. A neural probabilistic language model [J]. Journal of machine learning research, 2003, 3 (Feb): 1137-1155.

[13] Mikolov T, Le Q V, Sutskever I. Exploiting Similarities among Languages for Machine Translation [J/OL]. arXiv preprint arXiv, 2013: 1309 [2013-9-17]. https://arxiv.org/abs/1309.4168.

[14] Mikolov T, Chen K, Corrado G, Dean J. Efficient Estimation of Word Representations in Vector Space [J/OL]. arXiv preprint arXiv, 2013: 1301 [2013-9-7]. https://arxiv.org/abs/1301.3781.

[15] Zhang W T. Research and Application of Synonym Expansion Based on Feature Space Optimization of Word Vector Model [D]. Beijing University of Posts and Telecommunications, 2014.

[16] Zhu X M. Weibo recommendation based on Word2Vec topic extraction [D]. Beijing Institute of Technology, 2014.

[17] Tang M, Zhu L, Zou X C. A Document Vector Representation Based on Word2Vec [J]. Computer Science, 2016, 43 (6): 214-217.

[18] Wang Y, Liu Z, Sun M. Incorporating linguistic knowledge for learning distributed word representations [J]. PloS one, 2015, 10 (4): e0118437.

[19] Alsuhaibani M, Bollegala D, Maehara T, Kawarabayashi K. Jointly learning word embeddings using a corpus and a knowledge base [J]. PloS one, 2018, 13 (3): e0193094.

[20] Li Y, Wei B, Liu Y, et al. Incorporating knowledge into neural network for text representation [J]. Expert Systems with Applications, 2018, 96: 103-114.

[21] Le Q V, Mikolov T. Distributed Representations of Sentences and Documents [J]. 2014, 4: II-1188.

[22] Dai A M, Olah C, Le Q V. Document Embedding with Paragraph Vectors [J/OL]. arXiv preprint arXiv, 2015: 1507 [2015-7-29]. https://arxiv.org/abs/1507.07998.

[23] Fisher G, Israni M, Robert Z. Exploring Optimizations to Paragraph Vectors [J]. https://web.stanford.edu/class/cs224n/reports/2760664.pdf

[24] Grzegorczyk K, Kurdziel M. Binary Paragraph Vectors [J/OL]. arXiv preprint arXiv, 2017: 1611 [2017-6-9]. https://arxiv.org/abs/1611.01116.

[25] Palangi H, Deng L, Shen Y, et al. Deep sentence embedding using long short-term memory networks: analysis and application to information retrieval [J]. IEEE/ACM Transactions on Audio Speech & Language Processing, 2015, 24 (4): 694-707.

[26] Kennedy J, Eberhart R. Particle Swarm Optimization. In: Proc IEEE International Conference on Neural Networks. Perth, Australia, 1995: 1942-1948.