A Word Elimination Strategy for Learning Document Representation

Ying Liu¹,², Tongzhou Zhao¹,², Yue Chai¹,² and Yiqi Jiang¹,²

¹Hubei Provincial Key Laboratory of Intelligent Robot, Wuhan Institute of Technology, Wuhan, China.
²School of Computer Science & Engineering, Wuhan Institute of Technology, Wuhan, China.
Email: yingtpr@foxmail.com

Abstract. Computing word vectors based on neural network has motivations on document representation. Word elimination can enhance the extract effective of the quality of valuable feature of a document. In this paper, we propose a model named PV-IDF to eliminate redundancy and refine features to improve the performance of the classify model, with which tokens that carry semantic information of a document. The results show that PV-IDF model achieves state-of-art performance, especially for short-length document representation.

1. Introduction

Distributed representation associates with each word in the vocabulary a distributed word feature vector[1] and overcomes the curse of dimensionality that is caused by large number of parameters to be learned even with the small size of vocabulary[2]. Word2Vec and Paragraph2Vec are two typical Distributed representation methods. Word2Vec greatly compresses the vector dimensions when considering the contextual semantic information of a word[3]. However, dimensionality reduction Word2Vec rarely considers the impact of the order between words on the expression of full text and the weighting of each word. Paragraph2Vec extend Word2Vec method to treat sentences, paragraphs or entire documents as a block and produce dense feature vector with lower dimensional space, and vectors of words and documents can be generated at the same time when model was trained. But this approach ignores the document token which carries the entire semantic information of a document. Doc2VecC, which is motivated by Word2Vec can sustain substantial amount of syntactic and semantic meanings of a phrase or a sentence[4]. Doc2VecC learns document representation through a corruption, which randomly removes words from a document during token learning, and has a better performance and less training time. The technique of random elimination is equivalent to downsizing documents to a scale with ignoring the contribution of strong characteristic words. Consequently, the representations will have poor performance, especially on short-length document. Therefore, we aim at studying document representation method with capability of preserving semantic and syntactic characteristics.

In this work, we propose PV-IDF model to make the word elimination more reasonable and maintain an impressive level of representation quality for short-length document. To evaluate our method, several experiments are conducted on document classification task and efficiency evaluation of training time.
2. Related Work and Notation

Here we introduce core technology and representative methods for document representation. Notations: Assuming a training corpus $D$ contains documents, in which each document $D_i$ has a sequence of words $w_i$ with length of $T$. Then a vocabulary $V$ of size $v$ is built from $D$, in which word $w_i$, word frequency $f_i$ and inverse document frequency value $r_i$ are arranged in rows. Then, the local context $c$ and document token $d$ are used to predict target word $t_i$ during training.

2.1. Negative Sampling (NEG)

PV-DM model is an extension of Word2Vec for documents, which contain an input layer, a projection layer and an output layer[5]. It defines the probability of observing target word $w_i$ in a document $D_i$ given its local context $c_i$ and document token $d$ as

$$p(w_i | c_i, d) = \sigma \left( U \left[ Wc_i + \frac{1}{T} Wd \right] \right)$$

(1)

In basic representation model of Word2Vec, cost of calculation in output layer through softmax is impractical. Thus Mikolov came up with a technique, negative sampling, that use the sum of target and negative words predicting loss as the final result in output layer[6]. Consequently, the definition of negative sampling shown in e.q(2). Where $l$ is the label of predicted word, $l = 1$ indicates target word and $l = 0$ indicates negative samples.

$$L(W, c_i, d) = - \sum_{i=1}^{k} \left[ l \log p(w_i | c_i, d) + (1 - l) \log(1 - p(w_i | c_i, d)) \right]$$

(2)

2.2. Doc2VecC Model

Chen presented a document representation model named Document Vector through Corruption (Doc2VecC), the architecture was shown in figure 1. The main difference between Doc2VecC and PV-DM is the processing way of document token. Doc2VecC removes words randomly to change the document token into global context.

Figure 1. The architecture of Doc2VecC model.

The global context was defined as following. Initializing document representation $x$ and changing it into global context $\tilde{x}$ by randomly removing words from a document with probability $q$, formulated by e.q(3).

$$\tilde{x}_d = \begin{cases} 0, & \text{with possibility } q \\ x_d, & \text{otherwise} \\ 1-q, & \\ \end{cases}$$

(3)

Thus according to e.q(3), the objective function shown as e.q(4).
\[ L(W, c, x) = -\sum_{t=1}^{k} \left[ I \log \sigma \left( U \left( W_{c_t} + \frac{1}{T} W_{\bar{x}} \right) \right) + (1-I) \log \left( 1 - \sigma \left( U \left( W_{c_t} + \frac{1}{T} W_{\bar{x}} \right) \right) \right) \right] \] (4)

As reported, Doc2VecC achieves higher efficiency and quality for document representation than the work of Kiros and Mikolov[7]. But it is unreasonable that only uses random probability \( q \) to be words removing threshold. If we set \( q = 0.9 \) (according to Chen’s work) for 10 words document, then 9 words will remained to generate token. It is unclear whether these words are related to document topic. As the length of document decreases, the fewer words remained, which result in missing feature of document. Thus Doc2VecC can be deficient for short-length document.

3. Method

3.1. Document Tokens Based on Feature Words
Removing words from document to get tokens can reduce model training time obviously. However the criterion of word elimination can influence the quality of tokens. The random elimination strategy of Doc2VecC result in loss information of significance words, which inspired us to draw an ideal that only use feature words to form document tokens.

Inverse Document Frequency (IDF) is a measure of word importance. The more representative the word is, the lager IDF value it has. Each word in vocabulary has its own IDF value \( r_i \) and defines as e.q(5).

\[ r_i = \log \frac{|D|}{1 + N_i} \] (5)

where \( |D| \) is the size of training corpus, \( N_i \) is the number of documents where contain word \( w_i \). The words appear infrequently will have high IDF, which are potential to be feature words. After filter process, the remained words represent document features and generate token at input layer. Namely, those words with high IDF enhance the feature of documents more or less. Thus even a few words remained, the token can still carry the refined semantic information which ensure high-quality representations of short-length documents.

3.2. Representation for Documents: PV-IDF
Our model is modified from PV-DM at input layer, named PV-IDF and shown in figure 2. Taking a document ‘The film met with considerable critics and public acclaim.’ as an example. We elaborate the details of training process in this section.

![Figure 2. The architecture of PV-IDF model.](image)

It can be seen that local context (met, with, critics, and) of target word (considerable) were chosen at input layer. The average vectors of feature words \( \{ w_{y_x}, w_{y}, w_y \} \) were considered as the document token \( d_t \), which is different from Doc2VecC. And each unit in hidden layer has ability to calculate the average value. At output layer, negative sampling was used to provide the prediction of target word. Here a document token \( d_t \), which is average of feature word vectors, and objective function were described following by formulaic expressions.
In virtue of $r_i$, each feature word ($r_i > \theta$) will be remained. Thus document token of $w_i$ is given by eq.(6). It then defines the output $h$ of hidden layer as eq.(7). As for hidden layer, we use length of document $T$ as a reduced scale factor to normalized result. Consequently, the objective function $L(l, f)$ we used is cross entropy and given by eq.(8). According to NEG, $l$ is the label of target word, so objective function can be converted to eq.(9) by using property of sigmoid function.

$$d_i = \text{average}(w_x + w_y + w_z)$$  \hspace{1cm} (6)

$$h = Wc_i + \frac{1}{T}Wd_i$$  \hspace{1cm} (7)

$$L(l, f) = -\sum_{i=1}^{k+1} [l \log f(w_i | c_i, d_i) + (1-l)\log(1-f(w_i | c_i, d_i))]$$  \hspace{1cm} (8)

$$L(l, f) = -\log \sigma(Uh) - \sum_{j=1}^{4} \log \sigma(-Uh)$$  \hspace{1cm} (9)

Partially, many NLP tasks can be solved as long as we feed document vectors directly into optional machine learning algorithms.

4. Experiment
To prove the effectiveness and practicability of our work, several methods were used as baselines of which description shown in section 2. The experiments were conducted on desktop with Intel i7, 3.4GHz×8 cpu.

4.1. Datasets
The details of datasets covering these tasks were enumerated in table 1. Google snippets contain 8 domains with 10,060 for training and 2,280 for testing[8]; Hotel Reviews from Public comment network are positive and negative; OHSUMED includes medical abstracts from the Medical Subject Headings (MeSH) categories of the year 1991[9]; Sougou Chinese news contains 18 news channels from June 2012 to July; Fudan Corpus from Fudan Natural Language Processing Group is a collection of news from 20 domains. Imdb is a movie reviews collected by Stanford AI Lab[10]. Another movie reviews corpus is collected by Cornell[11].

| Dataset          | Google Snippets | Hotel Reviews | OHSUMED | Imdb  | Sougou Corpus | Cornell’s Reviews | Fudan Corpus |
|------------------|-----------------|---------------|---------|-------|---------------|-------------------|--------------|
| Size (Mb)        | 0.4             | 1.9           | 6.5     | 136   | 8.7           | 7.6               | 8.4          |
| Word Count       | 19              | 26            | 193     | 237   | 289           | 668               | 996          |

4.2. Baselines
PV-DM, NBSVM on n-gram and Doc2VecC were used as baselines. Here, n-gram features refers to uni-gram, bi-gram and tri-gram.

4.3. Task1: Document Classification
4.3.1. Setup. For experiment (1), we use Doc2VecC and PV-IDF on binary classification task. For experiment (2), we use all baselines and PV-IDF on 4 different datasets, in which each dataset has different length of train documents. Apart from that, keeping the same state with experiment (1). After generation of document vectors, we use logistic regression classifier for classification task through liblinear toolkit [12].
4.3.2. Result. Four considerable indicators were used in experiment (1), we can see PV-IDF has higher indicator values than Doc2VecC does from table 2. Then, the result of experiment (2) is shown in table 3. We can see PV-IDF performs better on various datasets in most cases, according to word count, the error rate gap between PV-IDF and methods (NBSVM+bi-gram and Doc2VecC) on datasets (Google Snippets and Hotel Reviews) is larger than others (OHSUMED and Fudan Corpus). That is to say, PV-IDF is more suitable for short-length documents.

Table 2. Five indicator results of Doc2VecC and PV-IDF on datasets.

| Dataset          | Word Count | Method      | TPR | PPV | F | MCC  |
|------------------|------------|-------------|-----|-----|----|------|
| Fudan Corpus     | 996        | Doc2VecC    | 0.97| 0.99| 0.98| 0.96 |
|                  |            | PV-IDF      | 0.98| 0.99| 0.99| 0.97 |
| Cornell’s Reviews| 668        | Doc2VecC    | 0.82| 0.88| 0.85| 0.71 |
|                  |            | PV-IDF      | 0.78| 0.87| 0.82| 0.66 |
| Imdb             | 237        | Doc2VecC    | 0.87| 0.87| 0.87| 0.74 |
|                  |            | PV-IDF      | 0.89| 0.89| 0.89| 0.78 |
| OHSUMED          | 193        | Doc2VecC    | 0.91| 0.92| 0.91| 0.83 |
|                  |            | PV-IDF      | 0.93| 0.93| 0.93| 0.87 |
| Hotel Reviews    | 26         | Doc2VecC    | 0.85| 0.84| 0.85| 0.69 |
|                  |            | PV-IDF      | 0.93| 0.92| 0.92| 0.84 |
| Google Snippets  | 19         | Doc2VecC    | 0.94| 0.95| 0.95| 0.89 |
|                  |            | PV-IDF      | 0.96| 0.99| 0.98| 0.95 |

Table 3. Comparison of error rate of various models on 4 datasets.

| Dataset            | Word Count | NBSVM+uni-gram | NBSVM+bi-gram | NBSVM+tri-gram | PV-DM | Doc2VecC | PV-IDF |
|--------------------|------------|----------------|---------------|----------------|-------|----------|--------|
| Google Snippets    | 19         | 4.3            | 12.7          | 4.3            | 5.0   | 5.1      | 2.0    |
| Hotel Reviews      | 26         | 17.5           | 13.5          | 10.0           | 11.7  | 10.5     | 7.2    |
| OHSUMED            | 193        | 6.0            | 8.8           | 9.5            | 9.9   | 8.4      | 6.7    |
| Fudan Corpus       | 996        | 5.1            | 3.7           | 8.2            | 2.2   | 1.9      | 1.3    |

4.4. Task2: Model Efficiency Evaluation

4.4.1. Setup. We trained 3 models on two different sizes dataset. And the evaluation criteria is the time of model learning and document vectors generation.

Table 4. Training and generation time required by models on Imdb and Cornell’s reviews

| Model   | Imdb            | Cornell’s Reviews |
|---------|-----------------|-------------------|
|         | Learning Time(s) | Generation Time(s) | Learning Time(s) | Generation Time(s) |
| PV-DM   | 1293            | 1270              | 149              | 129              |
| Doc2VecC| 760             | 8                 | 128              | 0.3              |
| PV-IDF  | 640             | 7                 | 103              | 0.3              |

4.4.2. Result. Table 4 summarized the time required by these models when they were training on two different corpora. PV-IDF is the fastest one to train, and Doc2VecC is the second. The length of required time for model training depends on the number of document in training corpus, which decides the count of parameters that need to be back-propagated in each update. PV-DM uses documents as features, which result in more parameters need to be update, while PV-IDF uses partial words. In spite of strategy for word elimination, generation time required by Doc2VecC and PV-IDF is nearly the
same. However learning time of PV-IDF is shorter, which can be firmly believed that the document token composed of feature words is beneficial to achieve acceleration of model training.

5. Conclusions and Future Work
We proposed a modified model with word elimination strategy for document features enhancement. Our experiments demonstrate PV-IDF is competitive with state-of-art methods. It has simple structure, and achieves significantly better performance than baselines in most cases. The document token in our model only contain a few feature words, however it outperforms impressively. As all baselines were declared have applicability for variable-length document representation. We also found our model performs even better for document of short-length, expressly comparing to Doc2VecC. Moreover, PV-IDF can alleviate cost brought by growing size of training corpus, and our strategy of word elimination enable the model to run on cheap hardware.

Nevertheless, there are still some work need to be done. We did not spend much time on adjusting hyper-parameters of our model. Although adjusting parameters can not fundamentally change the model effect, it is beneficial to the improvement of training efficiency. Actually, we used same parameter setting as Chen’s and achieved acceptable results. Another question worth exploring is how appropriate the threshold needs to be set to avoid omission of feature words.

6. Acknowledgments
This work was supported by the Graduate Education Innovation Fund(No. CX2017076).

7. References
[1] Bengio Y, Ducharme R, Vincent P, and Jauvin C 2003 A neural probabilistic language model Journal of machine learning research 3 Feb p 1137-1155
[2] Huang E H, Socher R, Manning C D, and Ng A Y 2012 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 p 873-882
[3] Mikolov T, Corrado G, Chen K, Dean J, Mikolov T, and Corrado G 2013 Efficient Estimation of Word Representations in Vector Space International Conference on Learning Representations p 1-12
[4] Chen M 2017 Efficient vector representation for documents through corruption arXiv preprint arXiv:1707.02377
[5] Le Q, and Mikolov T 2014 Distributed representations of sentences and documents International Conference on International Conference on Machine Learning vol 4 p II-1188
[6] Mikolov T, Sutskever I, Chen K, Corrado G, and Dean J 2013 Distributed representations of words and phrases and their compositionality International Conference on Neural Information Processing Systems vol 26 p 3111-3119
[7] Chen M 2017 Efficient vector representation for documents through corruption arXiv preprint arXiv:1707.02377
[8] Phan X H, Nguyen L M, and Horiguchi S 2008 Learning to classify short and sparse text & web with hidden topics from large-scale data collections In Proceedings of the 17th international conference on World Wide Web p 91-100
[9] Joachims T 1998 Text categorization with support vector machines: Learning with many relevant features In European conference on machine learning (Berlin:Springer) p 137-142
[10] Maas A L, Daly R E, Pham P T, Huang D, Ng A Y, and Potts C 2011 Learning word vectors for sentiment analysis In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies vol 1 p 142-150
[11] Pang B, and Lee L 2004 A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts In Proceedings of the 42nd annual meeting on Association for Computational Linguistics p 271
[12] Fan R E, Chang K W, Hsieh C J, Wang X R, and Lin C J 2008 LIBLINEAR: A library for large linear classification Journal of machine learning research p 1871-1874