CSLT TECHNICAL REPORT-20160033 [Sunday 8th January, 2017]

Memoryless Document Vector

Dongxu Zhang[1,3*] and Dong Wang[1,2]

*Correspondence:
wqx@cslt.riit.tsinghua.edu.cn
[1]Center for Speech and Language Technology, Research Institute of Information Technology, Tsinghua University, ROOM 1-303, BLDG FIT, 100084 Beijing, China
[3]CIST, Beijing University of Posts and Telecommunications , Beijing, China
Full list of author information is available at the end of the article

Abstract

We present a simple yet effective method to produce document vectors. Unlike conventional methods that allocate free parameters (or memory) to represent documents in model training, our method derives document vectors from word vectors by simple average pooling and therefore is memoryless. This enforces all semantic information of the training data being distributed and represented by words, leading to word vectors that are more semantic-loaded and more consistent to the average pooling-based document representation. Experiments on several topic and sentiment classification tasks show that our model outperforms the naive average pooling method, the PV-DBOW proposed by [1] and some other representative bag-of-words models.

Keywords: Word vector; DBOW

1 Introduction

Document representation plays a central role in a variety of tasks, such as document classification and text retrieval. The conventional TF-IDF method represents documents as high-dimensional vectors where each dimension corresponds to a particular word. An implicit assumption here is that words are independent of each other in semantics.

A better approach that can utilize the semantic relation among words is to represent words in a low-dimensional dense space, and then derive document vectors based on this low-dimensional word vectors. Many methods such as LSI [2] and the Paragraph Vector (PV) model [1] fall in this category.

A potential problem of these low-dimensional representation methods is the inconsistency between training and prediction. In training, word and document vectors are inferred simultaneously, while in prediction, word vectors are frozen and only document vectors are inferred. An assumption behind this word-fixed inference is that the semantic content of each word has been fully learned in the model training, and so documents can be simply derived from the semantic-loaded words. However, this is not necessarily true in practice, since the document vectors may absorb a significant proportion of semantic content of words in *model training*, due to the simultaneous word and document vector inference (e.g., by co-occurrence matrix factorization). To avoid such word semantic leakage, the document vectors should not be 'materialized' but derive from word vectors. We call this kind of document vectors 'memoryless', since they are computed instead of stored as free parameters when training the model.

A simple memoryless approach was proposed by [3], where the authors used the popular skip-gram model [4] to learn word vectors, and then used a simple average pooling to derive document vectors. Since no concept of document involved in

model training, all the semantic content of the training data is distributed to and represented by word vectors. This approach, however, does not involve any average pooling in model training, so the word vectors learned in this way are not necessarily suitable for average pooling-based document vectors.

To solve the potential problems mentioned above, we propose a memoryless document vector (MLDV) model that involves average pooling to infer document vectors when training word vectors. More specifically, the model optimizes word vectors such that the averaged vector (document vector) can well reconstruct the vectors of words that the document consists. The model structure is shown in Figure 1.

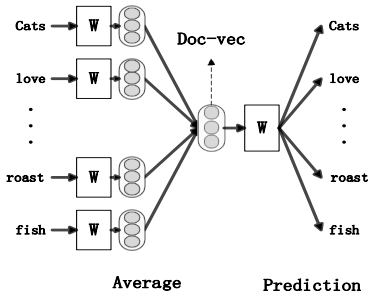

Figure 1 The training process of the proposed model.

The rest of the paper is organized as follows: Section 2 compares the proposed model to some related models, Section 3 presents the details of the model, and Section 4 reports the experimental results of four document classification tasks. Section 5 makes the conclusion.

2 Related Work

The most popular document vector approach is based on word distributions, e.g., TF-IDF. The vector it produces is large in dimensionality and considers no semantic relations between words. Topic models such as Latent Semantic Analysis (LSA) [2] tackle this issue by collaborating words and documents via the word-document co-occurrence matrix. Important topics are firstly discovered and then documents are represented by these topics. Latent Dirichlet Allocation (LDA) [5], Replicated Softmax [6] and DocNADE [7] all fall in this category.

Another prominent approach to document representation is the PV-DBOW model [1], which represents each document by a dense vector that is optimized so that the document words can be well predicted by this vector. A more sophisticated version of PV-DBOW is the PV-DM model, which leverages word sequence information. The PV-DM model generally outperforms PV-DBOW, though we focus on bag-of-words methods in this study so consider PV-DBOW only.

Most of the methods mentioned above share the same characteristic that documents are represented as free parameters. For instance, in LSA and LDA, documents are represented by distributions on topics, and in PV-DBOW, documents are represented as globally shared vectors. These document vectors hold significant semantic content of the training data, but are generally thrown away in prediction. In other words, some semantic information that should be represented by word vectors

now is 'memorized' by document vectors but is simply discarded at runtime. This is certainly not ideal. Our MLDV model does not memorize anything in document vectors; instead, all the semantic content of the training data is represented by word vectors.

[3] used average pooling to derive memoryless document vectors and reported surprisingly good performance on document classification tasks. Our model differs from their work in that the average pooling is involved in both model training (word embedding) and prediction (document vector derivation), which leads to word vectors that are naturally consistent to the pooling-based document vector derivation.

Our method is also similar to a combination of the skip-gram model and the cbow model [4], i.e., the average pooling of the word vectors in a local window is used to predict all the words in the same window. In our approach, the window covers the entire document, so the average pooling is directly connected to the word embedding and the document representation.

3 Model

The MLDV model structure has been shown in Figure 1. The basic principle is that all the words in a document are projected to a set of word vectors, and the average pooling result of these word vectors represents the document, in the sense that the pooled vector can well reconstruct the vectors of the words consisted in the document.

More specifically, the MLDV model involves two components: document vector derivation and word vector reconstruction. In the first component, the document vector is derived for a given document $d \in D$ that contains $L^{(d)}$ words. Each word token $d_i \in d$ is mapped to a low dimensional word vector W_{d_i} by looking up a word embedding matrix W. A simple average pooling is then applied to derive the document vector $v^{(d)}$, as formulated by:

$$v^{(d)} = (\sum_{i \in L^{(d)}} W_{d_i})/L^{(d)}. \tag{1}$$

In the second component, the document vector $v^{(d)}$ is used to predict all the words $\{d_i\}$ in document d. This idea is similar to the auto-encoder model, and assumes that a document vector should involve most of the semantic information of the words in the document. To formalize this idea, we maximize the conditional probability of all the document words $\{d_i \in d\}$ given $v^{(d)}$. The conditional probability of word d_i conditioned on the document vector $v_{(d)}$ is computed by a softmax function, given by:

$$\mathcal{P}(d_i|v^{(d)}) = \frac{W_{d_i}^T v^{(d)}}{\sum_j W_j^T v^{(d)}}, \tag{2}$$

where j indexes all the words in the vocabulary. The loss function to be minimized in model training is then given by:

$$\mathcal{L}(D; W) = \sum_{d \in D} \sum_{i \in L^{(d)}} -\log \mathcal{P}(d_i|v^{(d)}) \tag{3}$$

where the word vector matrix W contains the free parameters of the model. The stochastic gradient descent (SGD) algorithm is employed to train the model.

Once the word vectors have been fully trained, a document vector is simply derived by averaging the vectors of all the words in the document. Note that in both the model training and inference, only word vectors are trained and memorized, so the semantic information of a document is fully distributed to and represented by words.

4 Experiment

We test the proposed MLDV model on both topic classification and sentiment classification. A number of representative document representation approaches are implemented and compared to our model, including LSI [2], LDA [5], DocNADE [7], Skip-gram pooling [3], PV-DBOW [1]. Note that we focus on bag-of-words models, which are assumed to be simple and robust.

4.1 Datasets

Dataset	Statistics
Webkb	4 class; vocabulary size=7235; 2752 train data; 1370 test data
R8	8 class; vocabulary size=14477; 5449 train data; 2165 test data
20ng	20 class; vocabulary size=7298; 11218 train data; 7459 test data
SST	5 class; vocabulary size=17969; 9645 train data; 2206 test data

Table 1 Summary of the datasets used in the experiment.

Accuracy	R8		20ng		Webkb		SST	
	D=50	D=200	D=50	D=200	D=50	D=200	D=100	D=400
LSI	96.4	**97.2**	75.3	78.1	87.1	90.3	36.1	39.5
LDA	94.1	94.5	67.8	73.2	82.0	86.5	30.3	29.8
DocNADE	95.3	96.4	72.6	76.2	84.4	86.9	22.6	22.6
Skip-gram Pooling	96.3	96.5	75.4	78.1	86.4	86.8	37.1	38.7
PV-DBOW(our imp.)	96.1	95.3	75.2	76.2	**89.6**	90.0	34.0	36.8
MLDV	**96.5**	96.8	75.7	78.1	89.4	90.2	37.4	38.3
+initialization	96.0	96.7	**76.5**	**79.2**	89.3	**90.7**	**37.6**	**39.9**

Table 2 Accuracy on document classification with different bag-of-words models. 'D=50' denotes that the dimension of the document vector is 50.

Three datasets are used to conduct the topic classification experiments: Webkb, R8 and 20ng, as described by [8].[1] R8 is an 8-class dataset extracted from the Reuters21578 database, and 20ng is a 20-class dataset extracted from 20 news groups. Webkb is a 4-class dataset collected by the World Wide Knowledge Base project of the CMU text learning group. Stop words are removed using the SMART stop word list that consists of 571 words[2]. To limit the vocabulary size of the 20ng dataset, words whose frequency is lower than 20 are removed.

For the sentiment classification task, we use the 5-class Stanford Sentiment Treebank Dataset (SST) [9, 10]. The features of the four datasets are summarized in Table 1.

[1]The datasets are publicly available at `http://web.ist.utl.pt/acardoso/datasets/`

[2]`http://www.lextek.com/manuals/onix/stopwords2.html`

4.2 Baselines

Latent Semantic Indexing: The LSI model is implemented using the gensim tool[3] with the default configuration.

Latent Dirichlet Allocation: The LDA model is implemented using the gibbsLDA++ toolkit[4]. The parameter setting is: $iter$=2000, $arpha$=50/$topic\_size$, $beta$=0.01.

DocNADE: The DocNADE model is implemented using the tool published by [7].[5] We set $learning\_rate$=0.01, $epochs$=1000.

Skip-gram Pooling: The word2vec tool provided by Google[6] is used to train the skip-gram model. We choose the skip-gram model with hierarchical softmax, and set $epochs$=50, $window\_size$=10.

PV-DBOW: We implement the PV-DBOW model by ourselves. For both the training and test, we use $learning\_rate$=0.001, $epochs$=200.

4.3 MLDV model

When training the proposed MLDV model, the configuration is set to $learning\_rate$=0.001, $epochs$=200s. We also find that word vectors learned by the conventional skip-gram model can be used to initialize the MLDV model to achieve improved performance. This can be attributed to the local information that has been learned by the skip-gram model with a fixed window.

4.4 Classifier

Logistic regression (LR) is used as the classifier to evaluate performance on the document classification tasks. We use the scikit-learning toolkit[7] to implement the LR model, where the hyper-parameter C is tuned independently for each document representation approach. The optimal value is $C = 1$ for DocNADE, PV-DBOW and MLDV, and $C = 100$ for other baselines.

4.5 Results

The experimental results are presented in Table 2, where the performance is evaluated in terms of classification accuracy. For the topic classification task, we experiment two configurations of the document vector, one is 50 dimensional and the other is 200 dimensional; and for the sentiment classification task, the two configurations are 100 and 400 dimensions respectively.

From the results in Table 2, we can see that in most scenarios, the proposed MLDV model achieves the best performance. Another observation is that performance can be consistently improved by initializing the model with word vectors trained with a fixed window. These results support our conjecture that memoryless document vectors are more appropriate for document representation compared to conventional parameterized document vectors.

---

[3] http://radimrehurek.com/gensim/

[4] http://gibbslda.sourceforge.net/

[5] http://www.dmi.usherb.ca/~larocheh/code/DocNADE.zip

[6] http://code.google.com/p/word2vec/

[7] http://scikit-learn.org/stable/index.html

5 Conclusion

We proposed a simple memoryless document representation approach that involves average pooling when learning word vectors. The experiments on four document classification tasks demonstrate that the proposed model outperforms several representative baseline methods based on bag-of-word models. Future work involves integrating sequence information in the model, similar to PV-DM.

Author details
[1]Center for Speech and Language Technology, Research Institute of Information Technology, Tsinghua University, ROOM 1-303, BLDG FIT, 100084 Beijing, China. [2]Center for Speech and Language Technologies, Division of Technical Innovation and Development, Tsinghua National Laboratory for Information Science and Technology, ROOM 1-303, BLDG FIT, 100084 Beijing, China. [3]CIST, Beijing University of Posts and Telecommunications , Beijing, China. [4]Huilan LTD , Beijing, China.

References

1. Quoc Le and Tomas Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1188–1196.
2. S.C. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, and R.A. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society of Information Science*, vol. 41(6), pp. 391–407, 1990.
3. Chao Xing, Dong Wang, Xuewei Zhang, and Chao Liu, "Document classification based on i-vector distributions," in *APSIPA 2014*, 2014.
4. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.
5. David M Blei, Andrew Y Ng, and Michael I Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.
6. Geoffrey E. Hinton and Ruslan R Salakhutdinov, "Replicated softmax: an undirected topic model," in *Advances in Neural Information Processing Systems 22*, 2009, pp. 1607–1614.
7. Hugo Larochelle and Stanislas Lauly, "A neural autoregressive topic model," in *Advances in Neural Information Processing Systems*, 2012, pp. 2708–2716.
8. Ana Cardoso-Cachopo, "Improving methods for single-label text categorization," PdD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa, 2007.
9. Bo Pang and Lillian Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 2005, pp. 115–124.
10. Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*. Citeseer, 2013, vol. 1631, p. 1642.