Doc2Sent2Vec: A Novel Two-Phase Approach for Learning Document Representation

Ganesh J
IIIT, Hyderabad, India
ganesh.j@research.iiit.ac.in

Manish Gupta
Microsoft, India
gmanish@microsoft.com

Vasudeva Varma
IIIT, Hyderabad, India
vv@iiit.ac.in

ABSTRACT

Doc2Sent2Vec is an unsupervised approach to learn low-dimensional feature vector (or embedding) for a document. This embedding captures the semantics of the document and can be fed as input to machine learning algorithms to solve a myriad number of applications in the field of data mining and information retrieval. Some of these applications include document classification, retrieval, and ranking.

The proposed approach is two-phased. In the first phase, the model learns a vector for each sentence in the document using a standard word-level language model. In the next phase, it learns the document representation from the sentence sequence using a novel sentence-level language model. Intuitively, the first phase captures the word-level coherence to learn sentence embeddings, while the second phase captures the sentence-level coherence to learn document embeddings. Compared to the state-of-the-art models that learn document vectors directly from the word sequences, we hypothesize that the proposed decoupled strategy of learning sentence embeddings followed by document embeddings helps the model learn accurate and rich document representations.

We evaluate the learned document embeddings by considering two classification tasks: scientific article classification and Wikipedia page classification. Our model outperforms the current state-of-the-art models in the scientific article classification task by $\sim$12.07% and the Wikipedia page classification task by $\sim$6.93%, both in terms of F_1 score. These results highlight the superior quality of document embeddings learned by the Doc2Sent2Vec approach.

1. INTRODUCION

Document representations plays a vital role in the performance of several downstream IR applications such as document classification (or tagging), retrieval, ranking and so on. The most commonly used document representation is bag-of-words (BOW) or bag-of-n-grams [1]. Despite its simplicity and efficiency, it fails to capture the semantics of the documents as it suffers from data sparsity and curse of high dimensionality. Latent Dirichlet Allocation (LDA) [2] is another widely adopted distributed document representation.

In an attempt to harness the power of neural networks for document representations, Le et al. [3] propose a simple approach to learn document embedding from the word sequence using a standard word-level language model. The representations learned capture the ordering of words (unlike BOW) and also the semantics of the words in an efficient way (unlike LDA). In their follow-up work [4], the authors proposed an incremental model by jointly learning the word embeddings along with its document embedding. This change leads to learning rich and accurate representation compared to the previous model, which freezes the word vectors while learning the document vectors.

Inspired by the superior results obtained by the neural language models, we present a two-phase approach, Doc2Sent2Vec, to learn document embedding. In the first phase, we learn the sentence embedding using the word sequence generated from the sentence. Intuitively, the sentence representation is computed by modeling word-level coherence. In the next phase, we propose a novel model that learns the document representation from the sentence sequence generated from the document. Intuitively, the document embedding is computed by modeling sentence-level coherence. We argue in this work that the proposed decoupled strategy allows our model to compute accurate and rich document representations.

We validate the learned document embeddings using two classification tasks. In the first task, we aim at classifying a research article (or paper) among one of the eight different fields of computer science domain. In the second task, we predict the tag of a Wikipedia page. Doc2Sent2Vec outperforms the existing state-of-the-art model in scientific article classification task by $\sim$12.07% and Wikipedia page classification task by $\sim$6.93%, both in terms of F_1 score.

Our main contributions are summarized below.

- We present a novel two-phase approach, Doc2Sent2Vec, to learn document embeddings. To this end, we introduce a novel sentence-level language model which effectively constructs document representations by explicitly modeling sentence-level coherence.
- Experiments on Citation Network and Wikipedia datasets show that Doc2Sent2Vec learns high quality document embeddings outperforming the competitive baselines in both the classification tasks.

2. THE DOC2SENT2VEC APPROACH

SIGIR '16, July 17-21, 2016, Pisa, Italy

DOI: http://dx.doi.org/10.1145/2911451.2914717

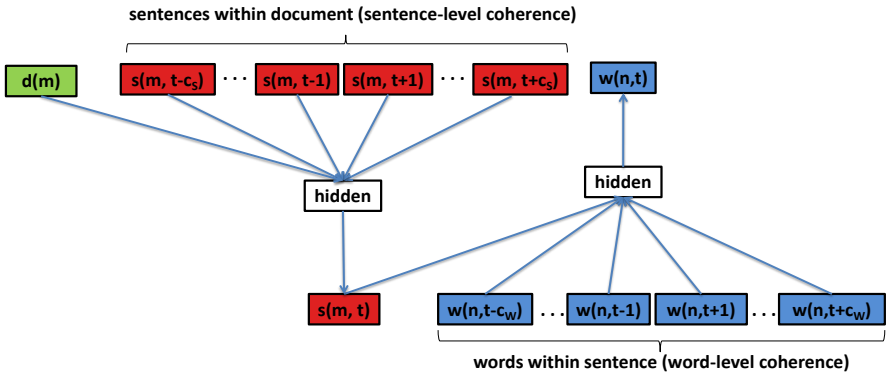

Figure 1: Architecture diagram of the Doc2Sent2Vec approach. Sentence embedding weights are shared between the two tiers.

Our hierarchical framework as shown in Figure 1 consists of two tiers; one learning the sentence representation from the word context, another learning the document representation from the sentence context.

Problem Formulation: Formally, let us denote a set D of M documents, $D = \{d_1, d_2, \cdots, d_M\}$, where each document d_m is a sequence of T_m sentences, $d_m = \{s_{(m,1)}, s_{(m,2)}, \cdots, s_{(m,T_m)}\}$. Each sentence is a sequence of T_n words, $s_{(m,n)} = \{w_{(m,n,1)}, w_{(m,n,2)}, \cdots, w_{(m,n,T_n)}\}$. For brevity, we drop the sentence index m when it is obvious in the context. The goal of the Doc2Sent2Vec approach is to jointly learn low-dimensional representations of words, sentences and documents as a continuous feature vector of dimensionality D_w, D_s and D_d respectively. We will realize this goal in two phases, as discussed in the following.

Modeling word-level coherence: In the first phase, the model aims to learn sentence representation from the word sequence within the sentence. We add the sentence vector to the standard language model that predicts the next word given its context word. This sentence vector must capture the topics of the sentence in a compact form. Each word is mapped to a unique vector, denoted by a column in the matrix V_{word}, whose size is given by $D_w \times |V_w|$ (where $|V_w|$ is the vocabulary size). Similarly, each sentence in a document is mapped to a unique vector denoted by a column in the matrix V_{sent} with size $D_s \times |V_s|$, where $|V_s|$ is the number of unique sentences. The model uses the concatenation of word vectors of context words along with the sentence vector as features to predict the given word in a sentence.

Formally, consider $w_{(n,t-c_w)}, \cdots, w_{(n,t-1)}, w_{(n,t+1)}, \cdots, w_{(n,t+c_w)}$ as the context words for the target word $w_{(n,t)}$, appearing in the sentence $s_{(m,n)}$. The objective of the word-level language model is to maximize the log likelihood probability.

$$\mathcal{L}_{word} = \sum_{d_m \in D} [\sum_{s_{(m,n)} \in d_m} \log \mathbb{P}(s_{(m,n)} | w_{(n,1)}, \cdots, w_{(n,T_n)}) + \sum_{s_{(m,n)} \in d_m} \sum_{w_{(n,t)} \in s_{(m,n)}} \log \mathbb{P}(w_{(n,t)} | w_{(n,t-c_w)}, \cdots, w_{(n,t-1)}, w_{(n,t+1)}, \cdots, w_{(n,t+c_w)}, s_{(m,n)})] \quad (1)$$

Here $2 \times c_w$ denotes the length of the context for the word sequence. The probability of observing the central word $w_{(n,t)}$ given the context words and the sentence is defined using the following softmax function.

$$\mathbb{P}(w_{(n,t)} | w_{(n,t-c_w)}, \cdots, w_{(n,t-1)}, w_{(n,t+1)}, \cdots, w_{(n,t+c_w)}, s_{(m,n)}) = \frac{exp\left(\bar{\mathbf{v}}^{1T}_{word} \mathbf{v}'_{w_{(n,t)}}\right)}{\sum_{w=1}^{|V_w|} exp\left(\bar{\mathbf{v}}^{1T}_{word} \mathbf{v}'_{w}\right)} \quad (2)$$

where $\mathbf{v}'_{w_{(n,t)}}$ is the output representation of $w_{(n,t)}$ and $\bar{\mathbf{v}}^1_{word}$ is the concatenation of the input embeddings (ignoring the central term $w_{(n,t)}$), with dimensionality $2 \times c_w \times D_w + D_s$.

$$\bar{\mathbf{v}}^1_{word} = [\mathbf{v}_{s_{(m,n)}}; \mathbf{v}_{w_{(n,t-c_w)}}; \cdots; \mathbf{v}_{w_{(n,t-1)}}; \mathbf{v}_{w_{(n,t+1)}}; \cdots; \mathbf{v}_{w_{(n,t+c_w)}};] \quad (3)$$

Similarly, we define the probability of observing the sentence $s_{(m,n)}$, given the words present in it as follows.

$$\mathbb{P}(s_{(m,n)} | w_{(n,1)}, \cdots, w_{(n,T_n)}) = \frac{exp(\bar{\mathbf{v}}^{2T}_{word} \mathbf{v}'_{s_{(m,n)}})}{\sum_{s=1}^{|V_s|} exp(\bar{\mathbf{v}}^{2T}_{word} \mathbf{v}'_{s})} \quad (4)$$

where $\mathbf{v}'_{s_{(m,n)}}$ is the output representation of $s_{m,n}$ and $\bar{\mathbf{v}}^2_{word}$ is the concatenation of the input embedding of all the words present in the sentence $s_{m,n}$, with dimensionality $T_n \times D_w$.

$$\bar{\mathbf{v}}^2_{word} = [\mathbf{v}_{w_{(n,1)}}; \cdots; \mathbf{v}_{w_{(n,T_n)}};] \quad (5)$$

Modeling sentence-level coherence: In the next phase, we propose a novel language model which constructs the document representations from the sentence sequence present in the document. The novel task is to predict the current sentence using the embeddings of the surrounding sentences and the document embedding as features. We add the document vector in the input layer of this model, that captures the topics of the entire document in a compact form. Each document is mapped to a unique vector denoted by a column in the matrix V_{doc}, whose size is given by $D_d \times |V_d|$ (where $|V_d|$ is the number of unique documents).

Formally, consider $s_{(m,t-c_s)}, \cdots, s_{(m,t-1)}, s_{(m,t+1)}, \cdots, s_{(m,t+c_s)}$ as the context sentences for the target sentence $s_{(m,t)}$, appearing in the document d_m. The objective of our novel sentence-level language model is to maximize the following log likelihood probability.

Dataset	# Docs	Labels
CND	8000	information retrieval, data mining, artificial intelligence, machine learning and pattern recognition, natural language and speech, computer vision, distributed and parallel computing, human-computer interaction
Wiki10+	19740	wiki, art, reference, people, culture, books, design, politics, technology, psychology, interesting, wikipedia, research, religion, music, math, development, theory, philosophy, article, language, science, programming, history, software

Table 1: Dataset Details

$$\mathcal{L}_{sent} = \sum_{d_m \in D} [\log \mathbb{P}(d_m | s_{(m,1)}, \cdots, s_{(m,T_m)}) + \sum_{s_{(m,t)} \in d_m} \log \mathbb{P}(s_{(m,t)} | s_{(m,t-c_s)}, \cdots, s_{(m,t-1)}, s_{(m,t+1)}, \cdots, s_{(m,t+c_s)}, d_m)] \quad (6)$$

Here $2 \times c_s$ denotes the length of the context for the sentence sequence. The probability of observing the central sentence $s_{(m,t)}$ given the context sentences and the document is defined using the softmax function as given below.

$$\mathbb{P}(s_{(m,t)} | s_{(m,t-c_s)}, \cdots, s_{(m,t-1)}, s_{(m,t+1)}, \cdots, s_{(m,t+c_s)}, d_m) = \frac{exp(\bar{\mathbf{v}}^{1^T}_{sent} \mathbf{v}'_{s_{(m,t)}})}{\sum_{s=1}^{|V_s|} exp(\bar{\mathbf{v}}^T_{sent} \mathbf{v}'_s)} \quad (7)$$

where $\mathbf{v}'_{s_{(m,t)}}$ is the output representation of $s_{(m,t)}$ and $\bar{\mathbf{v}}^1_{sent}$ is the concatenation of the input embeddings (ignoring the central term $s_{(m,t)}$), with dimensionality $2 \times c_s \times D_s + D_d$.

$$\bar{\mathbf{v}}^1_{sent} = [\mathbf{v}_{d_m}; \mathbf{v}_{s_{(m,t-c_s)}}; \cdots; \mathbf{v}_{s_{(m,t-1)}}; \mathbf{v}_{s_{(m,t+1)}}; \cdots; \mathbf{v}_{s_{(m,t+c_s)}};] \quad (8)$$

Similarly, we define the probability of observing the document d_m given the sentences present in it as follows.

$$\mathbb{P}(d_m | s_{(m,1)}, \cdots, s_{(m,T_m)}) = \frac{exp(\bar{\mathbf{v}}^{2^T}_{sent} \mathbf{v}'_{d_m})}{\sum_{d=1}^{|V_d|} exp(\bar{\mathbf{v}}^{2^T}_{sent} \mathbf{v}'_d)} \quad (9)$$

where $\mathbf{v}'_{d_m}$ is the output representation of d_m and $\bar{\mathbf{v}}^2_{sent}$ is the concatenation of the input embedding of all the sentences present in the document d_m, with dimensionality $T_m \times D_s$.

$$\bar{\mathbf{v}}^2_{sent} = [\mathbf{v}_{s_{(m,1)}}; \cdots; \mathbf{v}_{s_{(m,T_m)}};] \quad (10)$$

Training details: The overall objective function of Doc2Sent2Vec is to maximize the log likelihood probability as follows.

$$\mathcal{L} = \mathcal{L}_{word} + \mathcal{L}_{sent} \quad (11)$$

We employ stochastic gradient descent to learn the parameters, where the gradients are obtained via backpropagation [12], with fixed learning rate of 0.1. However, it takes $O(V_w)$, $O(V_s)$, $O(V_s)$ and $O(V_d)$ to compute $\nabla \log \mathbb{P}$ from Equations 2, 4, 7 and 9, which is undesirable in practice. Hence, we use hierarchical softmax [6], to facilitate faster training. The testing phase was excluded as the embeddings for all the documents in the dataset are estimated during the training phase.

3. EXPERIMENTS

In this section, we present the experimental results to show the effectiveness of the learned document embeddings by considering two classification tasks.

Model	F_1
Paragraph2Vec w/o WT [3]	0.1275
Paragraph2Vec [4]	0.135
Doc2Sent2Vec w/o WT	0.1288
Doc2Sent2Vec	**0.1513**

Table 2: Classification Performance on CND Dataset

Dataset Description: The dataset details are displayed in Table 1. Citation Network Dataset (CND) [5] consists of a collection of research papers (along with abstracts) from different fields of computer science domain. Inspired by the recent work [13], we use only a sample of the original dataset to speed up the training process. We construct a dataset of 8000 papers by randomly sampling 1000 research papers from 8 different fields, as mentioned in Table 1. In the second task where we perform Wikipedia page classification, we make use of Wiki10+ dataset [14], which contains one or more social tags (along with the number of users who have annotated this tag) for each Wikipedia page retrieved from delicious.com. We find the most frequent 25 social tags and only keep those documents that contain any of these tags. It results in a collection of 19740 documents as shown in Table 1, with each document associated with the most voted social tag. For simplicity, we consider only the first paragraph of the Wikipedia article for learning the embeddings.

Experimental Setup: In all our experiments, we consider the following four models.

- Paragraph2Vec w/o WT [3]: Paragraph2Vec algorithm without **W**ord **T**raining (i.e. the word embedding matrix V_w is freezed during training).
- Paragraph2Vec [4]: Extension of [3] which allows joint training of both word and document vectors.
- Doc2Sent2Vec w/o WT: Model discussed in Section 2 without word training.
- Doc2Sent2Vec: Model discussed in Section 2.

To ensure fair comparison, we empirically set C_w, C_s, D_w, D_s, D_d to 5, 1, 100, 100, 100 respectively, for all the models. We lowercase all the words, remove those which occur less than 10 (15) times in the CND (Wiki10+) corpus. We use pre-trained Glove [11] word vectors trained successively on Wikipedia 2014[1] and Gigaword 5[2] corpus, to initialize the word embeddings (V_w). It is important to notice that we use a linear classifier (one-vs-rest logistic regression[3]) to do prediction. It can be argued that the model performance can be improved using non-linear models, but this falls out of scope of our goal. We use 5-fold cross-validation to report the model performance for both the tasks.

[1] http://dumps.wikimedia.org/enwiki/20140102/
[2] https://catalog.ldc.upenn.edu/LDC2011T07
[3] http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Model	F_1
Paragraph2Vec w/o WT [3]	0.0476
Paragraph2Vec [4]	0.0445
Doc2Sent2Vec w/o WT	0.0401
Doc2Sent2Vec	**0.0509**

Table 3: Classification Performance on Wiki10+ Dataset

Analysis on CND: Scientific article classification results are shown in Table 2. We observe that it is beneficial to learn word vectors too while training instead of merely using pre-initialised word vectors. On incorporating the learning of word vectors, the improvement of ∼5.88% and ∼17.47% in F_1 score for Paragraph2Vec and Doc2Sent2Vec respectively, justifies this claim. Moreover, we see that our model outperforms the state-of-the-art model by a significant margin of ∼12.07%. This is mainly because our model is able to exploit both word-level and sentence-level coherence to enrich the embeddings.

Analysis on Wiki10+: We present the Wikipedia page classification results in Table 3. It is interesting to see that learning the word vectors has a negative impact for Paragraph2Vec algorithm. This is shown by a decline in the F_1 score by ∼6.5%. We believe that the word vectors before training are semantically accurate as they are learned from the complete Wikipedia corpus. While training them again, the vectors tend to get distorted leading to poor results. It can be argued that the same trend should follow for our model when we learn the word vectors. However, the Doc2Sent2Vec results indicate that the F_1 improves by ∼26.93% on learning word vectors. This illustrates that the proposed strategy of jointly exploiting sentence level and word level coherence is insensitive to the distortions generated by word vectors, resulting in robust embeddings of the Wikipedia pages. The performance improvement of ∼6.93% over the best baselines in terms of F_1 score, highlights the superiority of the Doc2Sent2Vec approach.

4. RELATED WORK

Our work is inspired by the outburst of representation learning works using neural networks [6, 8, 9] to solve many natural language processing tasks. In this work, we focus on the important problem of capturing the semantics of the document in a machine-understandable format (or representation).

A recent work Paragraph2Vec [3] provides a simple solution to this problem by extending the popular algorithm 'Word2Vec' [6]. Their strategy of adding a memory vector to the input layer of the neural network exhibits significantly better results than commonly used representations such as LDA, BOW and so on. In the original work the authors freeze the word vectors, while in the extension [4] they let the gradients update the word vectors too, along with document vectors. This trick further improves the results in understanding longer documents such as research articles and Wikipedia pages. Our work is inspired by the superior results of these neural language models.

Similar to the spirit of Doc2Sent2Vec, Djuric et al. [7] propose a Hierarchical Document Vector (HDV) model to learn representations from a document stream. In the first phase, the model learns document representation from the word sequence (similar to [3]). In the next phase, the model enriches the representations further by exploiting the document sequence in the stream. The Doc2Sent2Vec approach is different from HDV because our model does not assume the existence of a document stream and HDV does not model sentences.

5. CONCLUSION

We proposed a novel two-phase approach Doc2Sent2Vec to learn document representations in an unsupervised fashion. To this end, we introduced a novel sentence-level language model which exploits the sentence sequence present in the document. We validated the document embeddings by considering two classification tasks. Our classification results indicate the superiority of the proposed approach, thereby constituting a step towards learning accurate and rich document representations.

In the future, we plan to extend the current approach to a general multi-phase approach where every phase corresponds to a logical sub-division of a document like words, sentences, paragraphs, subsections, sections and documents. Also, it will be interesting to investigate how the document embeddings that are learned through the Doc2Sent2Vec approach can be enhanced by considering the document sequence in a stream such as news click-through streams [7].

Acknowledgement

This work is supported by SIGIR Donald B. Crouch Travel Grant. The authors would like to thank NVIDIA for donating one Tesla K40 GPU card.

6. REFERENCES

[1] Harris, Z.: Distributional structure. Word, 10(23). (1954) 146–162
[2] Blei, D., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. In: JMLR. (2013)
[3] Le, Q., Mikolov, T.: Distributed Representations of Sentences and Documents. In: ICML. (2014) 1188–1196
[4] Dai, A.M., Olah, C., Le, Q.V., Corrado, G.S.: Document embedding with paragraph vectors. In: NIPS Deep Learning Workshop. (2014)
[5] Chakraborty, T., Sikdar, S., Tammana, V., Ganguly, N., Mukherjee, A.: Computer Science Fields as Ground-truth Communities: Their Impact, Rise and Fall. In: ASONAM. (2013) 426–433
[6] Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. In: ICLR Workshop. (2013) vol. abs/1301.3781
[7] Djuric, N., Wu, H., Radosavljevic, V., Grbovic, M., Bhamidipati, N.: Hierarchical Neural Language Models for Joint Representation of Streaming Documents and their Content. In: WWW. (2015) 248–255
[8] Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A Neural Probabilistic Language Model. In: JMLR. (2003) 1137–1155
[9] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural Language Processing (Almost) from Scratch. In: JMLR. (2011) 2493–2537
[10] Morin, F., Bengio, Y.: Hierarchical Probabilistic Neural Network Language Model. In: AISTATS. (2005) 246–252
[11] Pennington, J., Socher, R., Manning, C.D.: GloVe: Global Vectors for Word Representation. In: EMNLP. (2014) 1532–1543
[12] Rumelhart, D., Hinton, G., Williams, R.: Learning Representations by Back-propagating Errors. In: Nature. (1986) 533–536
[13] Dos Santos, C.N., Gatti, M.: Deep convolutional neural networks for sentiment analysis of short texts. In: COLING. (2014) 69–78
[14] Zubiaga, A.: Enhancing navigation on wikipedia with social tags. In: arxiv. (2012) vol. abs/1202.5469