CBOS: Continuos Bag of Sentences for Learning Sentence Embeddings

Ye Yuan
Information Systems Technology and Design
SUTD
Singapore
ye_yuan@sutd.edu.sg

Yue Zhang
Information Systems Technology and Design
SUTD
Singapore
yue_zhang@sutd.edu.sg

***Abstract*—There has been recent work learning distributed sentence representations, which utilise neighbouring sentences as context for learning the embedding of a sentence. The setting is reminiscent of training word embeddings, yet no work has reported a baseline using the same training objective as learning word vectors. We fill this gap by empirically investigating the use of a Continuous Bag-of-Word (CBOW) objective, predicting the current sentence using its context sentences. We name this method a Continuous Bag-of-Sentences (CBOS) method. Results on standard benchmark show that CBOS is a highly competitive baseline for training sentence embeddings, outperforming most existing methods for text similarity measurement.**

***Keywords*-NN; sentence embeddings; distributed sentence representation**

I. Introduction

Learning distributed representations of sentences is a challenging yet useful task, which has gained increasing research attention recently [1], [2], [3], [4], [5], [6], [7]. A simple baseline directly utilises word embeddings, taking the average of word vectors in a sentence as the vector representation of the sentence. This method has been proved a strong baseline for many NLP tasks [5], [8], [9], [10], [11]. However, it does not consider relations between sentences in a document level context, which can be a limitation to its representation power. Intuitively, adjacent seances typically belong to the same coherent context, hence having closer relationship and sharing larger common information compared to a pair of random sentences. As a result, a line of work has leveraged neighbouring sentences as contexts to learn the embedding of a sentence [5], [6], [10].

In particular, Kenter [5] set a training objective to explicitly distinguish a pair of neighbouring sentences from random sentence pairs, where a Siamese network is utilized to calculate the similiary of a sentence pair. Kenter [5] used the average of word embeddings as a sentence embedding, therefore training word embedding parameters when maximising the sentence-level training objective. Kiros [2] leverage LSTMs to combine word vectors into a sentence vector.

While existing methods used different training objectives, the basic idea behind is consistent, in utilising sentential context information to learn distributed *sentence* representations. This setting is highly reminiscent of the *word* objective of word embeddings, which is to leverage context information for learning *word* vector. However, no existing work has reported simple baselines for training sentence embeddings using word embedding techniques directly, which can be both efficient and accurate.

We empirically verify the usefulness of word embedding methods for training sentence embeddings, using a Continuous Bag-of-Word (CBOW; [12]) inspired training objective. In particular, we take word embedding as model parameters, following Kenter [5], calculating sentence vectors as the sum of embeddings of the words in the sentence. Taking a 2-sentence window as the context, we set a maximum log-likelihood training objective, predicting the target sentence based on its context sentences. The same optimisation techniques by CBOS is used to training our model. We name the model a Continuous Bag-of-Sentences (CBOS) method.

On the STS task of SemEval 12 - 15 [13], [14], [15], [16], CBOS gives highly competitive results compared to existing methods for sentence embedding.

II. Related Work

There has been much work learning sentence representations. Most existing methods train sentence embedding as a by-product of optimisation of a end training objective. A typical example is encoder-decoder, where the encoder is used to learn a sentence vector and the decoder is used to yield an output sentence. Li *et al.* [7] adopt a hierarchical neural autoencoder to encode and decode words and sentences, via LSTM to obtain the representation of sentences and documents. Kiros *et al.* [2] propose Skip-Thought vector to learn sentence representations by encoding the current sentence to a representation, and then decoding it into its context sentences. Bowman *et al.* [17] adopt variational autoencoder to embed sentences by using a single-layer LSTM for both the encoder and the decoder. Gan *et al.* [6] adopt CNN to encode the current sentence and LSTM as a decoder to reconstruct the input sentence or to predict a future sentence.

In addition to the above encoder-decoder based model, Wieting *et al.* [18] propose a universal sentence embeddings method, which uses LSTM to model sentences, and uses hingle loss to train the model. Mueller *et al.* [4] introduce Siamese recurent architectures to learn sentence similarity; Kalchbrenner *et al.* [19] adopt a convolutional architecture dubbed the Dynamic Convolutional Neural Network (DCNN) for the semantic modelling of sentences. Tai [20] introduces a generalization of the standard LSTM

978-1-5386-1981-0/17/$31.00

architecture to Tree-LSTM for representing sentences. Yin *et al.* [3] use attention-based convolutional neural network(ABCNN) to model sentence pairs. Hu *et al.* [1] use a series of convolutional neural network architectures to match sentences.

Another related approach is to use a "shallow" model, and our model is also belong to this method. Le *et al.* [21] extend word2vec to doc2vec to get the sentences and documents representation. Kenter *et al.* [5] propose a similar model named Siamese CBOW model. In Siamese CBOW, the target is composed of the context sentences and negtive sentences, and it measures sentence distance with cosine similarity. Hill *et al.* [10] propose fastsent model to obtain sentence embedding, which try to predict the words in context sentences based current sentence.

Our method is different from Siamese CBOW in the way context sentences are predicted - while Siamese CBOW uses a Siamese network to predicts whether a pair of sentences are in each others context individually, we take the standard CBOW setting, predicting context sentences simultaneously. In contrast of our model, fastsent predicts the words in context sentences, rather than the whole sentences themselves.

III. Method

A. Continuous Bag-of-Words

The CBOW model [12] obtains word vectors by predicting the current word based on its context words. Every word is projected to a unique fixed-length vector. Assuming the vocabulary size is V, the given sequence of training words is w_1,w_2,...,w_n. For current word w_i, CBOW mdoel apply the softmax function to compute the probability of it:

$$p\left(w_i|w_{context_i}\right) = \frac{\exp\left(w_i^T \cdot w_{context_i}\right)}{\sum_{j\neq i}^{V} \exp\left(w_j^T \cdot w_{context_i}\right)}, \quad (1)$$

where, $w_{context_i} = \frac{1}{|context_i|} \sum_{w_k \in w_{context_i}} w_k$. The loss function is this:

$$L = -\sum_{i=1}^{n} \sum_{j\neq i}^{V} \log p\left(w_i|W_{context_i}\right) \quad (2)$$

For the vocabulary size can be very large, in practise hierarchical softmax [12] or negtive sampling method [12] is adopted to improve the training speed. For negtive sampling, several words are seltected randomly to be the negtive samples. We can obtain the target words set $w_{target_i} = \{w_i, w_{neg_1}, w_{neg_2}, \ldots, w_{neg_m}\}$. For $w_j \in w_{target_i}$, we apply the softmax funciton to compute the probability $p\left(w_i|w_{context_i}\right)$:

$$p\left(w_i|w_{context_i}\right) = \frac{\exp\left(w_i^T \cdot w_{context_i}\right)}{\sum_{w_k \in w_{target_i}} \exp\left(w_k^T \cdot w_{context_i}\right)} \quad (3)$$

The framework of for CBOW with negtive sampling method is shown in Figure 1.

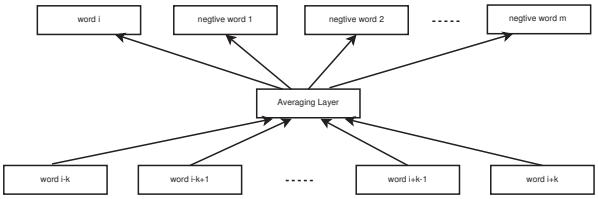

Figure 1. CBOW model architecture with negtive sampling.

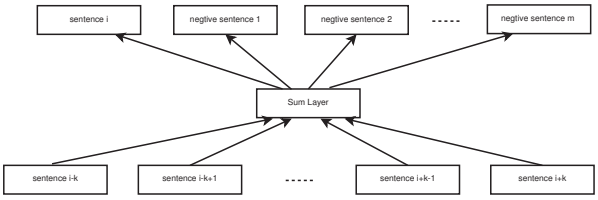

Figure 2. CBOS model architecture.

B. Continuous Bag-of-Sentences

Assume that the training data contains N sentences s_i and each sentence has T_i words represented with w_{ij}. We obtain an initial sentence state by adding up to the words' embedding.

$$s_i = \sum_{j=1}^{T_i} w_{ij} \quad (4)$$

We employ the sum of context sentences' embeddings around the current sentence s_i to represent the initial context state $s_{context_i}$:

$$s_{context_i} = \frac{1}{2k} \sum_{i-k\leq j\leq i+k, k\neq 0} s_j, \quad (5)$$

where k is the window size of the context. We predict s_i based on $s_{context_i}$, where the current sentence s_i can be seen as the positive target. Following CBOW, we randomly sample m sentences in the training data as negtive targets. The target sentences set $s_{target_i} = \{s_i, s_{neg_1}, s_{neg_2}, \ldots, s_{neg_m}\}$. For $s_j \in s_{target_i}$, we apply the softmax funciton to compute $p\left(s_j|s_{context_i}\right)$:

$$p\left(s_j|s_{context_i}\right) = \frac{\exp\left(s_j^T \cdot s_{context_i}\right)}{\sum_{s_k \in s_{target_i}} \exp\left(s_k^T \cdot s_{context_i}\right)} \quad (6)$$

The loss function of our model is defined below:

$$L = -\sum_{i=1}^{N} \sum_{s_j \in s_{target_i}} p\left(s_j\right) \cdot \log p\left(s_j|s_{context_i}\right), \quad (7)$$

where $p\left(s_j\right)$ denotes the target probability. If s_j is positive sample, the value will be 1.0, otherwise is 0.0.

The architecture of CBOS model as Figure 2 shows.

IV. Experiments

A. Baselines

Four models are used as our baselines.

- **CBOW** For the first baseline, we add up to the word embedding to represent the sentence from the CBOW model of word2vec [12].
- **Skip-gram** For the second baseline, we represent the sentence from Skip-gram model of word2vec [12] in the same way as the first baseline.
- **Skip-thought** We adopt skip-thought vectors [2] as our third baseline. The skip-thought vectors has been discribed in related work. It can directly model the sentence with LSTM to get the representation of the sentence.
- **Siamese CBOW** The siamese CBOW model proposed by Kenter [5] is as our forth baseline.

B. Training Dataset and Experiment Setup

The Toronto Book Corpus [2] consists all sentences in 11,038 books. It was used to train skip-thoughts [2]. We adopt it as the training dataset. We stem words with NLTK toolkit, and then get rid of special symbols, such as converting all words into lowercase letters. To make sure that all the sentence length is less than 120, we split those too long sentences to the short ones. And, we remove the words from dictionary whose frequencies are less than 5 when training the model.

We implement our model using TensorFlow and Python, training the model on NVIDIA GTX 1080 with batch sizes 10, number of iterations 10 and vector sizes 100.

C. Development Experiments

We study the influence of model's superparameter: the number of context sentences k and the number of negtive sentences m. The analysis is on the SemEvals Semantic Textual Similarity task 2014 [15]. We compute the cosine similarity between the sentence pair and adopt the Pearsons r to measure the gold-standard rating and the result.

1) Effect of context sentences' size k: This experiment studies the effect of the context sentences' size. We fix the number of negtive sentences to 2. Figure 3 shows the results with different the context sizes' k. In this figure, we can see that the value of k has different effects on various tasks. When the $k = 2$, the results of Deft-forum and headlines are the highest but those of Deft-news are the lowest. We can also see that the results are the highest when $k = 5$ in OnWN but the lowest in headlines and Tweet-news tasks. Considering time efficiency, we adopt the values $k = 1$.

2) Effect of negative sentences' size m: We fix the context size to $k = 1$. The results are shown in figure 4. We observe that the tasks OnWN, Deft-news, headlines and images reach the highest results when $m = 2$ or $m = 3$. While the task Def-forum and Tweet-news achive the best results when $m = 1$. We choose $m = 2$ for the size of negative sentences.

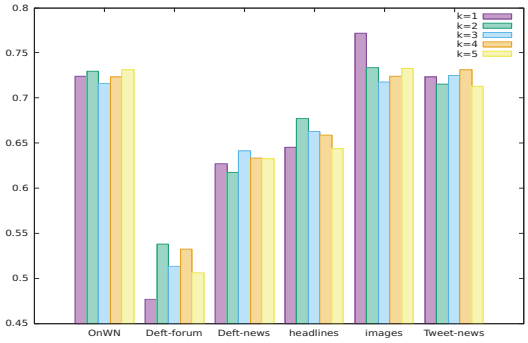

Figure 3. Effect of context sentences' size k.

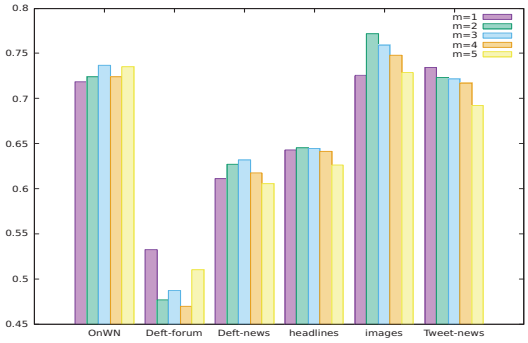

Figure 4. Effect of negative sentences' size m.

D. Final Results

In this section, we give the experiment result on the SemEval's Semantic Textual Similarity task 2012-2015 [13], [14], [15], [16], with $k = 1$ and $m = 2$.

Table 1 shows our results on the 20 tasks. We can see that our model gives the best results in 14 of all 20 tasks. We obtain more than 10% improvement in four tasks (14.47% in MSRvid 2012, 11.82% in OnWN 2013, 11.72% in OnWN 2014 and 12.25% in images 2014, respectly) compared to the baseline methods. Siamese CBOW [5] achieves best results in four tasks, with results by 0.01%, 0.48%, 0.75% and 2.13%, respectively, compared with our results, which is relatively close. Skip-thought and skip-gram obtain the best result in SMT 2013 and SMTnews 2012, respectly. The results show that our simple CBOS is highly useful in learning sentence representations.

V. Conclusion

We have investigated a simple baseline for training sentence embeddings, which leverages the continuous bag-of-word model for training word vectors. Similar to previous work, we use context sentences for deciding the embedding of a sentence. Different from existing state of the art methods, our training objective is directly taken from CBOW. Experiments show that the simple baseline performs competitively over a large benchmark of 20 datasets, demonstrating the effectiveness of continuous

Dataset	skipgram	CBOW	skip-thought	Siamese CBOW	CBOS
2012					
MSRpar	0.3740	0.3419	0.0560	0.4379	**0.4530**
MSRvid	0.5213	0.5099	0.5807	0.4522	**0.7254**
OnWN	0.6040	0.6320	0.6045	**0.6444**	0.6443
SMTeuroparl	0.3071	0.3976	0.4203	0.4503	**0.4599**
SMTnews	**0.4487**	0.4462	0.3911	0.3902	0.3967
average	0.451	0.4479	0.417	0.475	**0.5359**
2013					
FNWN	0.3480	0.2736	0.3124	0.2322	**0.3857**
OnWN	0.4745	0.5165	0.2418	0.4985	**0.6347**
SMT	0.1838	0.2494	**0.3378**	0.3312	0.3118
headlines	0.5935	0.5730	0.3861	**0.6534**	0.6486
average	0.40	0.4083	0.3195	0.4288	**0.4952**
2014					
OnWN	0.5848	0.6068	0.4682	0.6073	**0.7245**
deft-forum	0.3193	0.3339	0.3736	0.4082	**0.4764**
deft-news	0.5906	0.5737	0.4617	0.5913	**0.6269**
headlines	0.5790	0.5455	0.4031	0.6364	**0.6453**
images	0.5131	0.5056	0.4257	0.6497	**0.7722**
tweet-news	0.6336	0.6897	0.5138	**0.7315**	0.7233
average	0.5367	0.5425	0.4410	0.6041	**0.6614**
2015					
answ-forums	0.1892	0.1767	0.2784	0.2181	**0.3226**
answ-students	0.3233	0.3344	0.2661	**0.3671**	0.3458
belief	0.2435	0.3277	0.4584	0.4769	**0.5282**
headlines	0.1875	0.1806	0.1248	0.2151	**0.2186**
images	0.2454	0.2292	0.2100	0.2560	**0.2829**
average	0.2378	0.2497	0.26754	0.3066	**0.3396**

Table I
THE PERSON'S r RESULTS ON STS TASKS

bag-of-sentence training objectives for learning sentence vectors.

REFERENCES

[1] B. Hu et al., "Convolutional neural network architectures for matching natural language sentences," in *NIPS 2014*.

[2] R. Kiros et al., "Skip-thought vectors," in *NIPS 2015*.

[3] W. Yin et al., "Abcnn: Attention-based convolutional neural network for modeling sentence pairs," *TACL 2016*.

[4] J. Mueller et al., "Siamese recurrent architectures for learning sentence similarity." in *AAAI 2016*.

[5] T. Kenter et al., "Siamese cbow: Optimizing word embeddings for sentence representations," *ACL 2016*.

[6] Z. Gan et al., "Unsupervised learning of sentence representations using convolutional neural networks," *EMNLP 2017*.

[7] J. Li et al., "A hierarchical neural autoencoder for paragraphs and documents," *ACL 2015*.

[8] J. H. Lau and T. Baldwin, "An empirical evaluation of doc2vec with practical insights into document embedding generation," in *ACL 2016*.

[9] L. Yang et al., "Topical word embeddings," in *AAAI 2015*.

[10] F. Hill et al., "Learning distributed representations of sentences from unlabelled data," *NAACL 2016*.

[11] J. Matt et al., "From word embeddings to document distances," in *ICML 2015*.

[12] T. Mikolov et al., "Distributed representations of words and phrases and their compositionality," in *NIPS 2013*.

[13] E. Agirre and M. Diab et al., "Semeval-2012 task 6: A pilot on semantic textual similarity."

[14] E. Agirre and D. Cer et al., "Semval-2013 shared task: Semantic textual similarity, including a pilot on typed-similarity."

[15] E. Agirre and C. Banea et al., "Semeval-2014 task 10: Multilingual semantic textual similarity."

[16] E. Agirre and C. Baneab et al., "Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability."

[17] S. R. Bowman et al., "Generating sentences from a continuous space," *CoNLL 2014*.

[18] J. Wieting et al., "Towards universal paraphrastic sentence embeddings," *ICLR 2016*.

[19] N. Kalchbrenner et al., "A convolutional neural network for modelling sentences," in *ACL 2014*.

[20] K. S. Tai et al., "Improved semantic representations from tree-structured long short-term memory networks," in *ACL 2015*.

[21] Q. V. Le et al., "Distributed representations of sentences and documents." in *ICML 2014*.