Supervised Learning of Semantic Word Embeddings from Semantic Entailment Data

Ao Liu
College of Information and Computer Science
University of Massachusetts Amherst
Amherst, MA 01003
aoliu@cs.umass.edu

Abstract

Many natural language processing (NLP) tasks rely word embeddings. Traditionally, word embeddings are trained unsupervisedly on large corpus. This approach has in general worked well on many NLP related tasks. However, implementations of this unsupervised approach usually focus on the syntactic representations of words, thus lack of semantic interpretations. One of the problem is syntactic representations can hardly deal with negation and degree modifiers. Research shows that multiplicative embeddings perform better on semantic tasks, such as entailment. As an important task of semantics, entailment teaches the computer how to understand the meaning of a sentence and the implicit idea behind it. In this paper, I will show a supervised way to train multiplicative semantic word embeddings with the help of entailment dataset. Computer vision uses ImageNet to obtain features and then transfer them to other tasks. My work tends to indicate the capability of word embeddings trained on natural language inference to be transferred to other NLP tasks.

1 Introduction

Distributed representations of words (or word Embeddings)[1, 2, 3, 4] are generally used for many NLP tasks in recent years. Although these unsupervisedly trained word embeddings are capable of finding the general topic in a sentence based on the syntactic relationships of words, they still lack of semantic interpretations. Due to this problem, it is hard to find meaningful representation of a full sentence. A most recent paper shows that capturing the relationships among multiple words and phrases in a single vector to form the sentence representation can improve the performance on semantic entailment tasks [5]. However, the provided method using syntax-based word embeddings may still have ignored the impact of semantic meanings of words.

There are two major families of unsupervisedly learning wrod embeddings: 1) global matrix factorization methods, such as latent semantic analysis (LSA) [6] 2) co-occurrence methods, such as skip-gram model of Mikolov [3] and global log-bilinear regression model of Pennington *et al.* [4]. Pennington *et al.* seem better statistically solving global information, but the problem of semantics interpretation remain unsolved. Some significant examples are around 1) negation words, such as "not" and 2) degree modifiers, such as "more" and "less". Those examples are problematic because such negation words or degree modifiers can be added to any syntactically completed sentences, but the semantic meaning of the sentence may be altered for a certain degree. Training such words in an unsupervised manner is unwise and usually result in meaningless representations.

In my project, I try to study the task of learning semantic word embeddings that is trained on a large corpus in order to relieve some difficulties of finding more general semantic sentence representations. To obtain such semantic word embeddings, two major problem need to be solved, namely: what

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

would be a suitable neural network architecture; and how and on what task should such a network be trained. Most approaches of existing work on learning word embeddings in an unsupervised maner like word2vec [3] and GloVe [4]. I want to see if supervised learning can help solving the problem. The idea is inspired by previous result in NLP, where Alexis Conneau *et al.* train universal sentence representation [5], and in computer vision, where researchers pretrain models on the ImageNet [7] and transfer the learned features to other tasks. Conneau *et al.* show that training sentence encoders on a natural language inference (NLI) task achieve the best transferability [5]. The reason is that NLI is a high-level understanding task involving reasoning about the semantic relationships within sentences. Hence, I hypothesize that such task can also help training semantic word embeddings.

Unlike in computer vision, where convolutional neural networks (CNNs) dominate the field, many architectures are used in NLP tasks. Hence, I investigate different methods of training word embeddings with supervision, and compare their transferability to other task. My experiments show that the word embeddings recursively trained on the Stanford Natural Language Inference (SNLI) dataset [8] using the multiplicative trick yields better accuracies on SNIL dataset and the Microsoft Research Paraphrase Corpus than sentence-encoder-based method introduced by Conneau *et al* [5].

2 Related Work

Transfer learning using supervised features trained on ImageNet [7] has been successful on various computer vision tasks [9], such as face recognition [10] and visual question answering [11].

In contrast, most approaches of training word embeddings are unsupervised. This is mainly because people in NLP has not yet found the best supervised task for embedding the semantics of a word. With this reason, people compromise to unsupervised word embedding trained on large corpus, such as word2vec [3] and Gloval Vectors (GloVe) [4], which both statistically capture the syntactic co-occurrence information among words.

One aproach of training word embeddings in an unsupervised manner is to learn word representations using local context windows. Bengio *et al.* [1] introduced a model that learns word embeddings as a part of a neural network architecture for language modelling.

The skip-gram model of Mikolov *et al.* [3] propose a simple single-layer architecture based on the inner product between two word vectors. In their model, the objective is to predict a word's context given the word itself. The skip-gram model show its capability of learning linguistic patterns as linear relationships between word vectors with the evaluation on a word analogy task. Pennington *et al.* better solve the problem by including global co-occurrence statistics.

However, unsupervised learning models makes it harder for the word embeddings to specialize on semantic tasks. Conneau *et al.* [5] show that co-adaptation of sentence encoders and classifiers, when trained end-to-end on SNLI dataset, doesn't impact the generalization power of sentence features generated by an encoder.

To my knowledge, this work is the first attempt to use the SNLI dataset to train semantic word embeddings. As I show in my experiments, I can outperform unsupervised word embeddings and supervised sentence encoders on selected tasks.

3 Methodology

This work combines two research directions. First, I explain how the NLI task can be used to train semantic word embeddings using the SNLI task. Then I describe the architectures that I investigate for the embedding model and the entailment model. Specifically, for embedding model, I examine separated training of word embeddings and the end-to-end approach, and for entailment model I test different recurrent models such as gated recurrent units (GRU) [12], long short-term memory (LSTM) [13] and their bidirectional variants.

3.1 The Natural Language Inference Task

In SNLI, there are 570k human-written English sentence pairs labelled with one of the three categories: entailment, contradiction, and neutral. It captures NLI, or Recognizing Textual Entailment (RTE), and is one of the largest high-quality labelled dataset for semantics related NLP training purpose. I hypothesize that the semantic nature of NLI makes it a good candidate for learning semantic word embeddings in a supervised way. That is, I want to show that semantic word embeddings trained on NLI are able to capture the semantic meaning of words and can potentially improve the performance of entailment systems.

3.2 Word Embedding Models

There are two ways to train the word embeddings on SNLI: 1) adapt word co-occurrence statistics and 2) train embeddings of words in a sentence pair simultaneously.

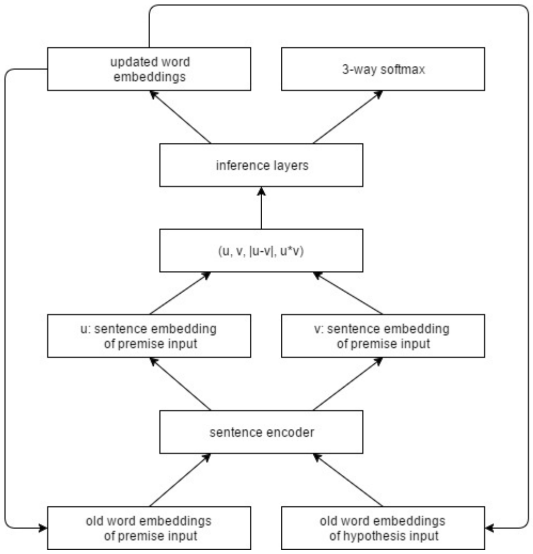
updated word embeddings
3-way softmax
inference layers
(u, v, |u-v|, u*v)
u: sentence embedding of premise input
v: sentence embedding of premise input
sentence encoder
old word embeddings of premise input
old word embeddings of hypothesis input

Figure 1: A generic architecture of learning word embeddings on SNLI in a supervised manner

Since training word embeddings within sentence pairs simultaneously implicitly shows the co-occurrences of words, I adopt the second setting. As illustrated in Figure 1, a typical architecture of this kind trains word embeddings of sentence pairs simultaneously. The word embeddings of premise and hypothesis inputs are passed through a shared sentence encoder to obtain the sentence embeddings premise u and hypothesis v. Once the sentence embeddings are generated, 3 matching methods are applied to extract relations between u and v: 1) concatenation of u and v, 2) element-wise absolute difference $|u - v|$ and 3) element-wise product $u * v$. The resulting vector will contain informations from both u and v and then be passed into inference layer(s) along with a softmax output layer. This will simultaneously update all the embeddings of words in the sentence pair and the updated word embeddings will then be applied to the following training iterations.

3.2.1 Pre-train Word Embeddings

In this setting, I hypothesize that sentence representation can be formed by the Cartesian point-wise multiplication of word embeddings. That is, a sentence $s = [w_1, ..., w_N]$ of length n has the vector representation v_s of the form $v_s = v_{w_1} \times ... \times v_{w_N}$, where v_{w_n} is the word embedding of n-th word in the sentence s. Thus, for a sentence pair, the division will result in 1 for entailment label and -1 for contradiction label, *i.e.* $u/v = 1$ for entailment and $u/v = -1$ for contradiction.

Then the model is only trained on sentence pairs with entailment and contradiction labels to obtain the pre-trained word embeddings.

3.2.2 End-to-End Approach

In this setting, I adopt the same hypothesis of embedding multiplication from the pre-trained model. In addition, the model is also trained on the traditional entailment architecture simultaneously so that the model jointly learns the word embeddings and the entailment model.

3.3 Entailment Model

Various neural networks for NLI exist. However, current systems are usually based on pre-trained word embeddings using unsupervised techniques. The architecture that better fits the semantic word embeddings is not yet clear. Thus, I compare different architectures: GRU- and LSTM- based models, and their bidirectional variants.

4 Datasets

I only choose two datasets. One is the SNLI dataset [8], where I train the word embeddings. The other one is the Microsoft Research (MSR) Paraphrase Corpus [14], where I evaluate the transferability of the word embeddings trained on the SNLI dataset. Data samples are shown in Table 1.

4.1 The SNLI Dataset

The SNLI dataset is a collection of 570k human-written English sentence pairs manually labeled for balanced classification with the labels entailment, contradiction, and neutral, supporting the task of natural language inference (NLI), also known as recognizing textual entailment (RTE).

4.2 The MSR Paraphrase Corpus

The MSR Paraphrase Corpus contains 5.8k sentence pairs obtained from a corpus of temporally and topically clustered news articles collected from thousands of web-based news sources using unsupervised techniques. For each sentence pair, the relation would be either true or false paraphrase.

Dataset	size	premise	hypothesis	label
SNLI	570k	A man inspects the uniform of a figure in some East Asian country.	The man is sleeping.	neutral
		An older and younger man smiling.	Two men are smiling and laughing at the cats playing on the floor.	neutral
		Some men are playing a sport.	Some men are playing a sport.	entailment
MSR Pharaphrase Corpus	5.8k	Ricky Clemons' brief, troubled Missouri basketball career is over.	Missouri kicked Ricky Clemons off its team, ending his troubled career there.	True
		Russ Britt is the Los Angeles Bureau Chief for CBS.MarketWatch.com.	Emily Church is London bureau chief of CBS.MarketWatch.com.	False

Table 1: **Data samples.** The SNLI dataset has labels of entailment, contradiction and neutral. The MSR Paraphrase Corpus only has labels of true and false paraphrase.

5 Experiments

My goal is to obtain general-purpose word embeddings that capture semantic information that can be transferred to other tasks. To evaluate the quality of the word embeddings trained on the SNLI dataset with the multiplication trick, I use them as features on MSR Paraphrase Corpus. I present my procedure of evaluating the word embeddings in this section.

5.1 Evaluation Method

Traditional evaluation on the quality of word embeddings are in three categories: 1) word analogy task, 2) word similarity task and 3) downstream NLP tasks, such as named entity recognition (NER). The word analogy task is to find the missing word b^* in the relation: a is to a^* as b is to b^*, where a, a^* are related by the same relation as b, b^*. A classic example is $king : man :: queen : woman$ [3]. Recent research show that solving this problem is equivalent to computing a linear combination of word similarities between the query word b^*, with the given word a, a^* and b [**?**]. However, Faruqui *et al.* found there is a problem with similarity-based evaluation of word embeddings [**?**]. That is, using word similarity task for evaluation of word embeddings may lead to incorrect inferences. Furthermore, they suggest task specific evaluation. For the purpose of finding better semantic word

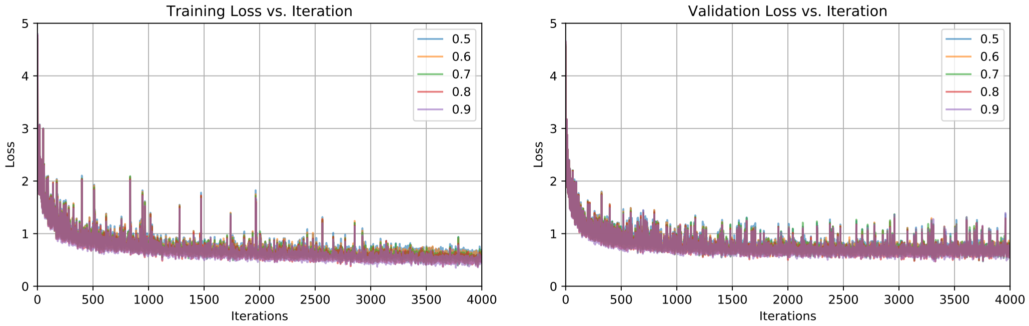

Figure 2: Training and validation loss on SNLI dataset.

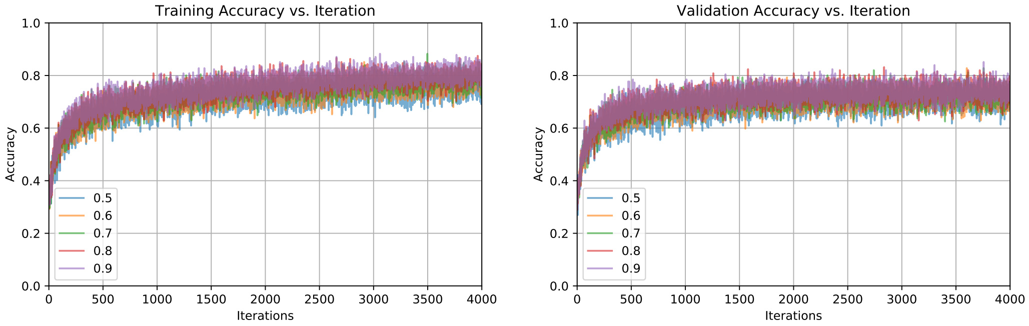

Figure 3: Training and validation acc on SNLI dataset.

embeddings, I evaluate the performance of the word embeddings on the original SNLI dataset and the MSR Paraphrase dataset. The preferred evaluation metric for both of tasks is accuracy, since the two tasks are either 3-way classification or binary classification problem.

5.2 Training Details

Since SNLI dataset has pre-split training, development and test datasets, I train my model on the training set, which contains 550k sentence pairs. I tokenize and lowercase the dataset with NLTK's word tokenizer, and build a vocabulary of 30k most frequent words.

For my experiments, I train the model using Adam optimization algorithm [**?**]. Since Adam using adaptive learning rate, I choose a reasonable starting learning rate, $3e-3$. I use the early-stop trick to test convergence, where there is no decrease in the loss of development set for the last 500 iterations. To compare with the performance of entailment system by Conneau *et al.* [5], I choose the same dimension of word embeddings as 300, which they adopt the pre-trained GloVe word embeddings, and the same number of hidden units of the inference layer as 512. I choose the mini-batch size as 256 for training. I use 512 hidden states for bi-LSTMs sentence encoder to save time for the purpose of course project. The dropout rate is chosen based on the performance on the development dataset of SNLI through cross-validation.

6 Results

6.1 Dropout Impact

To choose the best inverse dropout rate, I cross-validate the values of $[0.5, 0.6, 0.7, 0.8, 0.9]$. Figure 2 shows the training and validation losses during the training process, and Figure 3 shows the corresponding accuracies. The graph shows that when the inverse dropout rate is 0.9, I obtain the best trade-off between training loss and validation loss. And the model achieves the best accuracy of 85.2% on the development dataset of SNLI.

Model	dim	SNLI dev	SNLI test	MSR test
LSTM	2048	81.9	80.7	79.5
GRU	4096	82.4	81.8	81.7
BiGRU-last	4096	81.3	80.9	82.9
BiLSTM-Mean	4096	79.0	78.2	83.1
BiLSTM-Max	4096	85.0	84.5	**85.2**
BiLSTM(pre-trained embedding with multiplication trick) (**my model**)	**512**	83.6	83.1	83.4
BiLSTM with multiplication trick (**my model**)	**512**	**85.2**	**84.7**	**85.2**

Table 2: Accuracies of SNLI development and test datasets, and accuracies of the transfer task MSR Pharaphrase Corpus. The dimension column indicates the number of hidden units of the sentence encoder. The performances of the first five models are provided by Conneau *et al.* [5].

6.2 Transferability

From Table 2, we can see that my model, which uses Bi-LSTM of only 512 hidden units with multiplication trick, achieves the best accuracies of 85.2% and 84.7% on the development and test datasets of SNLI correspondingly. It also performs equally with the Bi-LSTM-Max model of Conneau *et al.* when transfers to MSR Paraphrase Corpus and gets accuracy of 85.2%.

7 Conclusions

This paper studies the effects of training semantic word embeddings with supervised data by transferring to another semantic task. My results shows that my word embedding model trained on SNLI dataset can perform better than models trained with pre-trained unsupervised word embeddings. With word embeddings trained using the multiplication trick, a simple Bi-LSTM sentence encoder outperforms the complex Bi-LSTM-Max sentence encoding method of Conneau *et al.* .

I believe that my work only scratches the surface of possible models for learning semantic word embeddings. Larger datasets of semantics could improve the quality of semantic word embeddings.

Future work may fall into two categories. One way is to test the performance of the trained word embeddings on other semantic tasks, such as Semantic Textual Similarity. The other is to find better model of training such embeddings. Due to the time we have for the course project, I did not explore much more other possible architectures.

References

[1] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[2] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[3] 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*, pages 3111–3119, 2013.

[4] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[5] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.

[6] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391, 1990.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[8] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

[9] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

[10] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[11] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015.

[12] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[14] Bill Dolan, Chris Quirk, and Chris Brockett. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics*, page 350. Association for Computational Linguistics, 2004.