Milestone: Supervised Learning of Multiplicative Semantic Word Embeddings from Semantic Entailment Data

Ao Liu
University of Massachusetts Amherst
140 Governors Dr., Amherst, MA 01003
aoliu@umass.edu

1. Introduction

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. Syntactic representations usually follow additive rules, i.e. "king" - "man" + "woman" = "queen". 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. I want to find a supervised way to train multiplicative semantic word embeddings with the help of paraphrase/entailment dataset(s). Due to the change on embeddings, I will also implement a new entailment model architecture better fitting the multiplicative property of the presented word embeddings.

2. Problem Statement

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 *et al*. [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 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 embeddings 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.

There are mainly two datasets that I may use. One is the Stanford Natural Language Inference (SNLI) corpus [8], which 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). The other is Paraphrase Database (PPDB) [9], which is an extensive semantic resource, consisting of a list of phrase pairs with (heuristic) confidence estimates. The dataset to be used may vary, depending on further exploration and understanding on them.

Unlike in computer vision, where convolutional neural networks (CNNs) dominate the field, many architectures are used in NLP tasks. I want to investigate the impact of various semantic entailment architectures using neural networks and compare their performances on top of the learned semantic word embeddings. The models will be evaluated on the accuracy of the entailment task, *i.e*. the percentage of correctly inferred relationship of sentence pairs in the dataset.

3. Technical Approach

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 entailment model. Specifically, I examine standard recurrent models such as LSTMs [**?**] and CNNs.

3.1. The Natural Language Inference Task

There are 570k human-written English sentence pairs labelled with one of the three categories: entailment, contradiction, and neutral. It captures NLI, or 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.

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.

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.

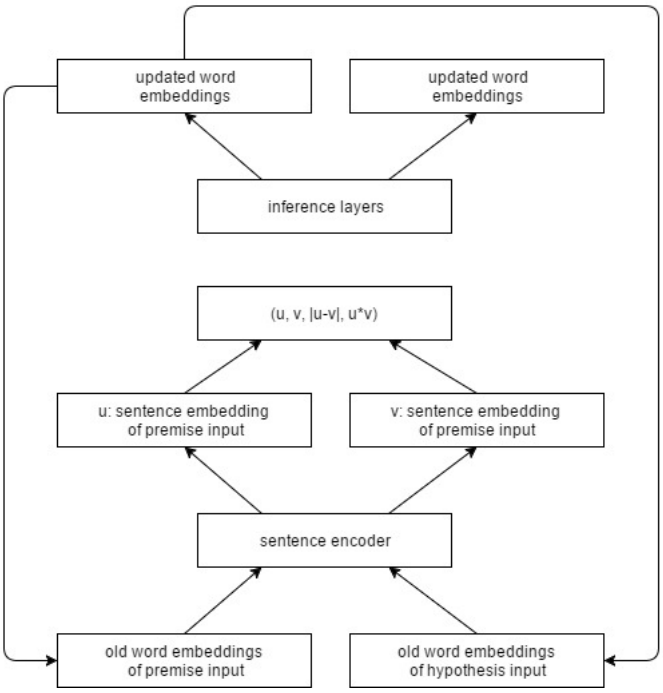

Figure 1. A generic architecture of learning word embeddings on SNLI in a supervised manner

3.2. Sentence Entailment Architectures

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. I want to compare different architectures: standard recurrent encoders with LSTMs or Bi-LSTMs, fully connected layers and even CNNs.

4. Intermediate Result

Conneau *et al*. public their PyTorch implementation of their sentence encoder on GitHub [5]. For convenience of my implementation, I reimplement their work in TensorFlow and achieve similar results on SNLI (See Table 1).

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 lan-

Model	dim	dev	test
LSTM	2048	81.9	80.7
GRU	4096	82.4	81.8
BiGRU-last	4096	81.3	80.9
BiLSTM-mean	4096	79.0	78.2
BiLSTM-Max	4096	85.0	84.5

Table 1. Performance of sentence encoder architectures on SNLI. Dimensions of embeddings were selected according to best aggregated scores.

guage 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] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. Ppdb: The paraphrase database. In *HLT-NAACL*, pages 758–764, 2013.