Word Embeddings for Fine-Grained Sentiment Analysis

Mohan Rao Divate Kodandarama
Department of Computer Science
University of Wisconsin-Madison

divatekodand@wisc.edu

Abstract

We introduce a novel approach to learn word embeddings which when used for sentiment recognition tasks improves the accuracy by noticeable degree.

1 Antonym-enhancing cosine loss

Word vectors can perform surprisingly well in representing semantic and syntactic relationships between words. For example, word vectors learned from current state-of-the-art approaches (e.g. [**?**]) can answer syntactic questions such as - "What is the word that is similar to 'big' in the same sense as 'smaller' is to 'small'". However, such vector representations only capture similarity of words as per their place in a corpus. In particular, they fail to capture the compositional semantics present in text. Consider the following two sentences:

The movie was excellent.

The movie was terrible.

The first sentence has a strong positive connotation, while the second sentence has a strong negative connotation. Current word vector representations represent antonyms such as 'excellent' and 'terrible' as very similar and place them close in the vector space. We hypothesize that a vectorization that takes into account these antonyms would be immensely beneficial for sentiment recognition systems relying on word vector representations.

To address this, we propose a novel word vector representation called the Cosine Loss Word Embeddings (CLWE), which tries to learn orthogonal vector representations for antonyms. More formally, the architecture of learning CLWE is similar to Continuous Skip-gram Model with negative sampling ([**?**]), but it additionally adds a cosine loss term for every pair of antonyms seen during the training phase. Continuous Skip-gram model tries to maximize the classification of a word based on another word in the same sentence. Given a sequence of training words $w_1, w_2, w_3, ..., w_T$, the objective of the Skip-gram model is to maximize the average log probability:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c<=j<=c, j\neq 0} log\, p(w_{t+j}|w_t)$$

where c is the size of the training context. The basic Skip-gram formulation defines $p(w_{t+j}|w_t)$ using the softmax function:

$$p(w_O|w_I) = \frac{exp(v_{w_O}^T v_{w_I})}{\sum_{w=1}^{W} exp(v_{w_O}^T v_{w_I})}$$

where v_{w_O} and v_{w_I} are input and output vector representations of w, and W is the number of words in the vocabulary. This formulation is impractical because of large cost of computing the gradient of the softmax. Hence an alternate approach called the Negative Sampling is often used. Negative Sampling Objective:

$$log\ \sigma(v_{w_O}^T v_{w_I}) + \sum_{i=1}^{k} E_{w_i \sim P_n(w)}[log\ \sigma(v_{w_i}^T v_{w_I})]$$

In CLWE, in addition to maximizing the negative sampling objective, we try to minimize the cosine similarity between the word vectors corresponding to antonyms.

$$L = \text{CosineLoss}(u, v) = \frac{u^T v}{||u||\ ||v||}$$

For each pair of antonyms seen during the training phase, we add the above additional loss term. This results in additional gradient term, described below for pairs of antonyms seen during the training phase.

$$\frac{\partial L}{\partial u_i} = \frac{v_i \sum u_i^2 - u_i\ u^T v}{(\sum u_i^2)^{\frac{3}{2}} \sqrt{\sum v_i^2}}$$

Where u and v are word vector representation for a pair of antonyms. In addition, a hyper-parameter β is use to tune the learning rate of the cosine loss term. Our experiments indicate that the values of β in the range 10^{-5} to 10^{-4} are useful. 1 shows that this approach learns orthogonal word vector representations for antonyms. Further, **??** shows the nearest neighbours of a given word, with respect to cosine similarity measure. It is seen that the nearest neighbour set of a word in cosine loss embedding does not contain any of its antonyms.

Table 1: Cosine simlarity between 300 dimension word vectors corresponding to common antonyms. Both the Models - Word2vec-Skipgram model and Cosine loss embeddings, were trained on a subset (40 Million) of 1 Billion Word Language Model Benchmark dataset

Antonym Pairs	Word2vec Skip-gram Model	Cosine loss embeddings
bad:good	0.17013	0.00151
ended:begin	0.10618	0.00115
short:long	0.24898	2.1166e-05
demonstrating:disprove	0.17387	0.04159
forgot:remember	0.17388	0.00542
Rejecting:accept	0.20799	0.00237
early:late	0.19553	0.01330
fall:ascent	0.14891	0.00211
contradict:affirm	0.14889	0.044182
Sells:buy	0.14847	0.06270

We used the a subset (40 Million words) of a 1 Billion Word Language Model Benchmark dataset (`http://www.statmt.org/lm-benchmark/1-billion-word-language-modeling-benchmark-r13output.tar.gz`) for learning all our word embeddings. In particular, we trained the standard word2vec skip-gram model with negative sampling and a CLWE on this dataset. Antonym pair information was mined using the WordNet®[**?**] lexical database. We evaluate the quality of each of the word vector representations by using them in several sentiment recognition models.

Table 2: Nearest neighbours of the word *good* in Embedding space

Word2vec Skip-gram model		Cosine loss embedding	
Nearest Neighbour	Cosine Similarity	Nearest Neighbour	Cosine Similarity
bad	0.717059	better	0.945991
lousy	0.666702	best	0.93051
decent	0.652319	well	0.78963
nice	0.646020	improving	0.56774
mediocre	0.582150	ameliorate	0.529369