Evaluation of Distributional Compositional Operations on Collocations through Semantic Similarity

Drozdova Ksenia

National Research University Higher School of Economics
Moscow, Russia
drozdova.xenia@gmail.com

Abstract. This paper considers comparative estimation of compositional distributional semantic models. Central to our approach is the idea that the meaning of a phrase is a function of the meanings of its parts. We provide two vector space models - for lemmatized and unlemmatized corpus, and four compositional functions, which we tested on a phrase similarity task. Our main goal is to estimate, which method most accurately expresses the relationship of whole

Keywords: compositional distributional semantic models, vector word representations, word2vec, semantic similarity

1 Introduction

This paper presents a comparative study of compositional distributional semantic models. The experiments have been inspired by Gottlob Frege's classical idea of compositionality, that is to say, the meaning of a phrase is a function of the meaning of its parts [1].

With the aid of neural language models it is possible to test this statement and select a function which would best express the connection between the whole and its parts regarding our data. This work also analyzes the question whether it is more effective to lemmatize a corpus prior to training a model or work with unlemmatized data.

In order to create a vector semantic space the author has used distributional semantics predictive algorithms that have been realized in the utility *word2vec* [2]. With the aid of these algorithms it is possible to create a vector space where words from the lexicon of the training corpus are put. First the coordinates of the words (their vectors) are initialized randomly, but during the process of training the similarity between the vectors of words that are neighbors in the corpus is maximized and the similarity between the vectors of the words that are not situated close to one another is minimized. The logic of such organization of space is based on the idea that words found in similar contexts usually have similar meanings and words whose contexts are not similar are semantically different. There is a metric of semantic similarity of words in this vector space;

the metric is defined as vector cosine similarity. Thus, using neural models it is possible see the semantic map of the language from the calculation viewpoint.

The utility *word2vec* has realized two training algorithms: Continuous skip-gram (skip-gram) and Continuous bag-of-words (CBOW). The model will be trained differently depending on the choice of algorithm. The objective function of the skip-gram algorithm is to predict a context using a word. The parameter window size defines what is considered a context; its value is equal to the maximum distance between the current word and the word that is being predicted. That is to say, the context of the word w_i with the window size k would be $w_{i-k}, ..., w_{i-1}, w_i, w_{i+1}, ..., w_{i+k}$. The objective function of CBOW is the opposite – it predicts a word using its nearest neighbors [3].

2 Description of the parameters of the models

We has chosen Russian National Corpus as the training corpus for creating the semantic space. The study has been conducted using two models one of which has been trained using a lemmatized corpus and the other has been trained using an unlemmatized corpus. Henceforth these models will be referred to as **Lemm** and **Token** respectively. In both models stop words have been filtered out and the same set of hyperparameters is used:

- dimensionality of the feature vectors is equal 300;
- window size 10;
- ignore all words with total frequency lower than 5;
- the training algorithm is skip-gram;
- negative sampling is used (5 samples);
- number of iterations over the corpus is equal 5.

In order to construct the models the author has used Gensim [6], particularly its module Phrases which detects common phrases and substitutes spaces between the words in such a phrase for underscores. For example, the common phrase '*Третий Рим*' ('*Third Rome'*) will be transformed into the token '*Третий_Рим*' ('*Third_Rome'*). Thus, the model will create vectors not only for separate word forms but also for collocations. Henceforth such vectors will be referred to as baseline.

3 Composition functions

Let us return to compositionality of phrases. In general we can describe the representation of a certain phrase w which consists of words $w_1, w_2, ...w_n$ as a vector $\overrightarrow{w} = \overrightarrow{w}_1 \star \overrightarrow{w}_2 \star ... \star \overrightarrow{w}_n$, where $\star$ can mean addition $+$, point-wise multiplication $\odot$, tensor product $\otimes$ and other operations on vectors.

Such composition functions have been described in detail in the work [5]. Its authors Jeff Mitchell and Mirella Lapata have researched methods based on multiplication of the corresponding elements and addition of vectors, countable

distributional semantic models and models that have been created with the aid of LDA. The work [4] studies the use of such compositional methods on prediction algorithms.

This paper considers four methods of creating a vector of a phrase using its components: a sum of vectors, element-wise multiplication, a weighted sum, tensor contraction, baseline (see Table 1). The weighted sum method supposes that phrase components should have different weights when added: the coefficient of the first component is α whereas the coefficient of the second component is $\beta = 1 - \alpha$. In order to evaluate which way represents the semantic map of the language best the quality of each model is calculated.

Method	**Function**	**Formula**
Addition	$\overrightarrow{p} = \overrightarrow{x} + \overrightarrow{y}$	$p_i = x_i + y_i$
Multiplication	$\overrightarrow{p} = \overrightarrow{x} \odot \overrightarrow{y}$	$p_i = x_i \cdot y_i$
Weighted Addition	$\overrightarrow{p} = \alpha\overrightarrow{x} + \beta\overrightarrow{y}$	$p_i = \alpha x_i + \beta y_i$
Circular Convolution	$\overrightarrow{p} = \overrightarrow{x} * \overrightarrow{y}$	$p_i = \sum_j x_j \cdot y_{i-j}$
Baseline	$p = x\_y$	$\overrightarrow{p}$ is produced by algorithm

Table 1: Compositional methods

The quality of the models is evaluated with the aid of Spearman's coefficient of correlation between a man's estimation of the semantic similarity of phrases – the so-called 'gold standard' [1] – and the cosine similarity of the vectors of the same phrases:

$$similarity(\overrightarrow{p_1}, \overrightarrow{p_2}) = \cos(\overrightarrow{p_1}, \overrightarrow{p_2}) = \frac{\overrightarrow{p_1} \cdot \overrightarrow{p_2}}{|\overrightarrow{p_1}||\overrightarrow{p_2}|} \quad (1)$$

The data of the gold standard consist mainly of phrases of the Adj+Noun type. In this experiment the author has used 105 pairs of phrases that the author has translated into the Russian language. The phrases have been lemmatized for the model **Lemm**.

4 Experiments results

During the experiments the author has calculated the optimum weights for the weighted sum: $\alpha = 0.6$ for the first phrase component, $\beta = 1 - \alpha = 0.4$ for the second phrase component. This can be seen on the graph 4 where the horizontal axis is for the quality of the model and the vertical axis is for the value of the α parameter.

[1] `http://adapt.seiee.sjtu.edu.cn/similarity/SimCompleteResults.pdf`

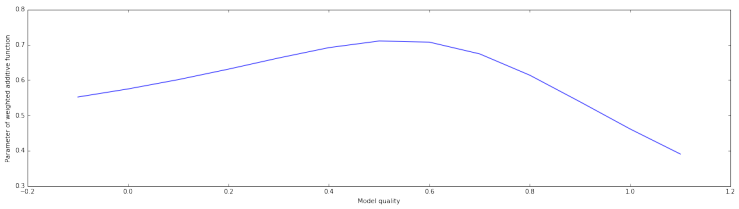

Fig. 1. The dependence of the model quality on parameters for weighted addition model

The table 2 contains the results of the experiment – the values of Spearman's coefficient of correlation between the methods the author has studied and the gold standard.

Method	**Lemm**	**Token**
Addition	0.71157	0.53716
Multiplication	0.27132	0.22719
Weighted Addition	**0.71335**	0.54661
Circular Convolution	0.07431	0.07931
Baseline	0.70766	0.53014

Table 2: Spearman ρ correlations of models with human judgements

As can be seen in the table, the best result belongs to the **weighted sum method** used on a lemmatized corpus. Element-wise multiplication and tensor contraction did not produce good results which can be easily shown using geometrical representation: these ways suppose that a new vector can be placed randomly relative to its components in the vector space which means that the basic characteristics of semantic space are not preserved.

Despite the fact that the best result of creating a vector for a phrase belongs to the weighted sum method, baseline has proved to be a very good way. The values of the correlation coefficients belonging to the best method and baseline differ by only 0.01.

It is interesting that the model trained on the unlemmatized corpus has produced much worse results that the model trained on the lemmatized corpus. However, it should be noted that the methods of weighted sum, baseline and simple addition have proved the most effective on both models.

5 Conclusion

This paper describes experiments with forming collocations that have been conducted with the aid of distributional semantic models. The study has two aims: to find out whether the model creates semantic space of the language better with lemmatization or without it and to determine which of the four compositional methods the author has described in this paper is the most effective in terms of creating vectors of phrases.

The most important result is that the neural language model forms a better vector representation for a lemmatized corpus, and the difference in results compared to those of an unlemmatized corpus is quite considerable (about 20 percent).

The question whether one should use compositional methods when working with collocations or turn to natural baseline (to unite collocations into a token before training) should be studied further engaging more data that would include combinations of various parts of speech. This paper has shown that the quality of baseline can be considered equal to the quality one could receive using the best compositional operator. It has been firmly determined by the research that this operator is the weighted sum of vectors:

$$\overrightarrow{p} = 0.6\overrightarrow{x} + 0.4\overrightarrow{y} \tag{2}$$

References

1. Frege, G. On sense and reference. 1892. *Ludlow* (1997), 563–584.
2. Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
3. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (2013), pp. 3111–3119.
4. Milajevs, D., Kartsaklis, D., Sadrzadeh, M., and Purver, M. Evaluating neural word representations in tensor-based compositional settings. *arXiv preprint arXiv:1408.6179* (2014).
5. Mitchell, J., and Lapata, M. Composition in distributional models of semantics. *Cognitive science 34*, 8 (2010), 1388–1429.
6. Řehůřek, R., and Sojka, P. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* (Valletta, Malta, May 2010), ELRA, pp. 45–50.