

Clustering of Co-Occurring Neighboring Unambiguous Terms (COCONUT)

Anouk Visser

anouk.visser@student.uva.nl

Rémi de Zoeten

Cristina Gârbacea

cr1st1na.garbacea@gmail.com

Abstract

Vector space models for word representation have shown to be useful in capturing the relationships between words’ functions and meanings. Similarities between words are encoded under the form of distance or angle in a high dimensional space. Neural language models, although less used than the traditional n -gram models because of their notoriously long training times, present superior performance on the task of word prediction. Leaving from the work of (Mikolov et al., 2013), we propose three new methods for word sense disambiguation based on the co-occurrence frequency of the context words near a given target word. We show that these are valid approaches in an unsupervised setting and can increase the accuracy of capturing syntactic and semantic regularities for the English language.

1 Introduction

The introduction will be here.

2 Related Work

Recently (Mikolov et al., 2013) have shown that linguistic regularities in continuous space word representations can be identified by a vector offset method based on cosine distance. Pairs of words that share the same relationship are observed to present a constant vector offset, which enables the measurement of syntactic linguistic regularity inside a set of analogy questions of the form “ A is to B as C is to $___$ ”, and semantic linguistic regularity by determining the extent to which given two pairs of words $A : B$ and $C : D$, the semantic relations between A and B are similar to those between C and D . The recursive neural network model they

employ for this task is trained with backpropagation to maximize data likelihood and consists of one input layer that accepts one word at a time encoded using l -of- N encoding scheme, one output layer which outputs a probability distribution over possible words and one hidden layer with recurrent connections that keeps track of the sentence history. The embedding vectors x_A, x_B, x_C are used to determine the word which is assumed to be the best answer to a question, $y = x_B - x_A + x_C$, or in case there is no word in space at this position, the word having the greatest cosine similarity with y . In the case of semantic evaluation where D is given, computing $\cos(x_B - x_A + x_C, x_D)$ determines the measure of relational similarity between the prototypical and target word pairs.

3 COCONUT

For learning the word representations (Mikolov et al., 2013) train an RNN with co-occurrence vectors of words. Instead of representing words by just one co-occurrence vector, we propose to train the model with multiple co-occurrence vectors for ambiguous words. The meaning of the word ‘apple’ can be determined by looking at its surrounding words, which could be: technology, iPhone, company for ‘Apple’, the company or: fruit, orchard, pie for ‘apple’ the fruit. COCONUT assumes that the meaning of a word is highly dependent on the words that accompany it and that the co-occurring words that define one meaning of ‘apple’ are more likely to co-occur with each other than two words that define two different meanings of apple (‘iPhone’ and ‘technology’ are more likely to occur together than ‘iPhone’ and ‘orchard’). COCONUT will attempt to split the co-occurrence vector for ‘apple’ into two co-occurrence vectors, one containing ‘iPhone’, ‘technology’ and ‘company’, the other containing ‘fruit’, ‘orchard’ and ‘pie’.

3.1 Co-Occurrence Vectors

We construct the co-occurrence vector for word A by computing the relatedness of word A with every other word in the vocabulary. We use the same function for relatedness as (Guthrie et al., 1991):

$$r(x, y) = \frac{f_{xy}}{f_x + f_y - f_{xy}}$$

where f_{xy} denotes the frequency of x and y occurring together and f_x and f_y denote the frequency of x , respectively y .

3.2 Latent Semantic Analysis

Unsupervised word sense disambiguation approaches exploit the idea that similar senses of a word have similar neighboring words. They try to induce word senses from input text by clustering word co-occurrences, aiming to divide "the occurrences of a word into a number of classes by determining for any two occurrences whether they belong to the same sense or not" (Schutze et al., 1997). Computing a vector for each word inside a corpus conflates word senses.

3.3 Similarity Measures

Measuring the similarity between two vectors can be seen as an equivalent to measuring their distance. Inversion or subtraction can be easily applied to transform a measure of distance between vectors into a measure of similarity.

The most common way to measure similarity between two vectors is to compute the *cosine* of the angle between them as the inner product of the two vectors, after they have been normalized to unit length: $\cos(x, y) = \frac{x \cdot y}{\|x\| \cdot \|y\|}$. Hence the length of the vectors is irrelevant. (Bullinaria et al., 1997) show that the cosine is highly reliable and performs the best, after having compared it with distance measures like Hellinger, Bhattacharya, and Kullback-Leibler. Other common geometric metrics frequently used in the vector space are represented by the Euclidean, Manhattan and Mahalanobis distance, Dice, Jaccard, Pearson and Spearman correlation coefficients.

The *Euclidean* distance between two points is defined as the length of the line connecting them. In the vector space, it is defined as $d(p, q) = d(q, p) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$. The smaller this distance the more similar the objects are. In a similar manner, the *Manhattan* distance is defined as the sum of the absolute differences of the coordinates

of two given points as $d(p, q) = |\sum_{i=1}^n (p_i - q_i)|$. The *Mahalanobis* distance generalizes the standard Euclidean distance by modelling the relations of elements in different dimensions. Given two vectors x and y , their squared Mahalanobis distance is $d_A = (x - y)^T A (x - y)$, where A is a positive semidefinite matrix.

The *Pearson* correlation coefficient is defined in a similar manner with the Spearman correlation coefficient, with the mention that the last one is between the ranked variables:

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}, \text{ where } \bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

and $\bar{y} = \frac{\sum_{i=1}^n y_i}{n}$. A value of 1 indicates total positive correlation, i.e. that a "best fit" line with a positive slope is generated which runs through all the datapoints, a value of 0 means no correlation and a value of -1 represents negative correlation between the two variables. The main advantage of this method over the Euclidean distance is that it is more robust against data which is not normalized.

3.4 Clustering

To find the two senses of a word, we apply k-means clustering to the co-occurrence vectors of the co-occurring words. COCONUT assumes that the words assigned to each cluster represent a different meaning of a word. Words that are not closely related to A do not contribute to either one of the meanings. Therefore, we will not use the co-occurrence vectors of all co-occurring words, but only those from the words that are closely related. Building a good decision process for defining when a word is closely related to another word is beyond the scope of this project and will most likely not necessarily lead to significant performance improvements. Therefore, we have decided to discard the words that have a relatedness score with A that falls in the bottom 50% of all relatedness-scores. Let the set of words that remains be called C . We can use the co-occurrence vectors of the words in C to find clusters, but these vectors will contain a lot of words that are not in C , do not occur together with A or do occur with A but not in C . We are only interested in finding clusters representing the different meanings of word A , therefore we will only use the co-occurring words in the vectors of C that are present in C .

4 Evaluation

We have evaluated the performance of CO-CONUT on a dataset containing X unique words, and has size X . Initially, we decided not to disambiguate the top X words, after extracting the two senses of the words and their distance, we discarded half of the disambiguated words, leaving us with X words that were disambiguated.

4.1 Empirical Evaluation

4.2 Quantitative Evaluation

5 Conclusion

References

- Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. *Proceedings of NAACL-HLT*, 746–751
- Joe A. Guthrie, Louise Guthrie, Yorick Wilks and Homa Aidinejad. 1991. Subject-dependent co-occurrence and word sense disambiguation. *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, 146–152 Association for Computational Linguistics
- John Bullinaria and John Levy. 1997. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behaviour Research Methods*, 510–526
- Schutze H. 1998. Automatic Word Sense Discrimination. *Computational Linguistic*, 97