Clustering of Co-Occurring Neighboring Unambiguous Terms (COCONUT)

Anouk Visser anouk.visser@student.uva.nl
Rémi de Zoeten remi.de.z@gmail.com
Cristina Gârbacea c.1st1na.garbacea@gmail.com

Abstract

Let's decide on the abstract once we finish the body of the text.

1 Introduction

Recently (Mikolov et al., 2013a) found that continuous space word representations capture syntactic and semantic regularities. For example they find that *queen* $-$ *king* $\approx$ *woman* $-$ *man*. The authors use these linguistics regularities to answer a set of analogy questions in the form of 'a is to b as c is to ...'. In this framework, every word is represented by exactly one continuous space word representation. One potential problem when answering analogy questions is that words can be ambiguous. When the question is '*seed* is to *apple* as *window* is to *house*' then the word apple refers to a fruit. Another question could be '*apple* is to *computer* as *porsche* is to *cars*' where the company apple is implied. Clearly, these are two very distinct entities, but the approach that (Mikolov et al., 2013a) presented does not differentiate between the two different meanings of apple. We propose to disambiguate between the various senses of a word in order to obtain multiple continuous space words representations for one word.

We present two different methods for word-sense disambiguation. In section 3 we describe how word-sense disambiguation can be accomplished by local co-occurrence clustering. In section 4 we propose another method, COCONUT, that uses global co-occurrences to disambiguate words.

2 Related Work

The linguistic regularities in continuous space word representations used in (Mikolov et al., 2013a) can be identified by using a vector offset method based on cosine similarity. A recurrent neural network language model is used to obtain the continuous space word representations. The RNN is trained using backpropagation to maximize data likelihood and consists of one input layer that accepts one word at a time encoded using *1*-of-*N* encoding scheme, an output layer which outputs a probability distribution over possible words and a 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.

Linguistic regularities can be used in many different applications unsupervised language learning applications. An example application is presented by (Mikolov et al., 2013b), where a method is proposed to exploit similarities among languages for machine translation. For machine translation some sense of a dictionary or phrase table is required. However, these are not always available, or are incomplete. By using two monolingual corpora a model can be trained and linguistic regularities can be learned for the two different languages. If the dictionary entry for 'queen' is missing, but, for example, the entry for 'king' is available, we can find the translation for 'queen' by using the vector offset method as described above.

3 Word-sense disambiguation by local co-occurrence clustering

To find the two senses of a word, one approach would be to cluster different meanings of a word based on the words that it co-occurs with in the corpus. This is done by generating a co-occurrence vector for every time the word is ob-

served in the corpus (this is a local co-occurrence vector). A co-occurrence vector is derived by observing the context of a word. In our experiments the frequency of the words that fall within a window of 5 words from the word that is being observed are encoded into the vector. This means that each vector is a sparse vector with the length of the vocabulary size, but can be encoded with at most 10 terms. These co-occurrence vectors are then clustered using k-means clustering. It is possible (even likely) that two co-occurrence vectors have no word in common, but still end up in the same cluster. For example, in the case of apple the words {*technology, iphone, company, revenue*} might be in the same cluster. Given the co-occurrence vectors, 1:{*technology, iphone*}, 2:{*iphone, revenue*}, 3:{*technology, company*} and 4:{*company, revenue*} then 1 and 4 have nothing in common, but can still be bound together by 2 and 3. It should be noted that extracting all co-occurrence vectors from a corpus can can require a significant amount of memory, even when using sparse-vector encoding. However, it is possible to have a fine-tunable tradeoff between memory requirements and the number of loops over the corpus (which is more cpu-intensive), by only recording a specific subset of the vocabulary on each iteration.

4 COCONUT

The COCONUT method for disambiguating words is based on two assumptions:

1. the meaning of a word is highly dependent on the words accompanying it

2. the co-occurring words that define one meaning of a word are more likely to co-occur with each other than two words that define two different meanings of the word

For example, for the two different meanings of the word 'apple', 'iPhone' and 'technology' are more likely to occur together than 'iPhone' and 'baking'.

In COCONUT, the words that accompany the word that we want to disambiguate, A, are the words that co-occur with A. COCONUT will 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'.

Let C be the set of words that co-occur with A. COCONUT first constructs and converts the global co-occurrence vectors of the words in C to relatedness vectors. It will then cluster these relatedness vectors in order to determine the two possibly different meanings of A.

4.1 Co-Occurrence Vectors

A global co-occurrence vector is a vector that contains all words that co-occur with a given word and the frequency of the two words co-occurring together. We obtain the global co-occurrence vector in a similar way of obtaining the local co-occurrence vector as described in section 3. The only difference is that instead of maintaining all local co-occurrence vectors for a given word, we will accumulate all local co-occurrence vectors in one global co-occurrence vector.

After obtaining the global co-occurrence vectors for every word in the corpus, we construct the co-occurrence vector for word A by computing the relatedness of word A with every other word in the C. We use the same function for relatedness as (Guthrie et al., 1991):

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

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.

Words that are not closely related to A do not contribute to either one of the meanings. Therefore, we will discard the words that have a relatedness score with A that falls in the bottom 50% of all relatedness-scores from C. The terms that are discarded are considered relevant to all meanings of A, we will call this set R.

4.2 Clustering and splitting

Next, we extract the co-occurrence vectors from the remaining words in C, we will call this set of co-occurrence vectors V. After applying k-means clustering on the vectors in V we expect to find two cluster centers that represent the two meanings for A. Note that we are only interested in describing the two meanings of A using the words in C. Therefore, for every vector in V we will discard all words that are not in C. The adjusted vectors can now be used to perform k-means clustering.

The two new co-occurrence vectors for A are initialized with the words in R. As the cluster

centers define the different meanings of A, we can look at the words in each cluster to fill the new co-occurrence vectors for A. For example if the words 'technology', 'iPhone' and 'company' are assigned to one cluster, they will be inserted into one of the new co-occurrence vectors for the word 'apple' while the words 'fruit', 'pie', 'baking' that were assigned to the other cluster will be inserted into the other co-occurrence vector.

COCONUT will split every word in the corpus in order to find two different meanings (we excluded the 75 most frequent words), but not all words are ambiguous. We expect that words that have two distinct meanings will have a greater cluster distance (i.e. a greater distance between the two meanings) than words that do not. We discard all disambiguations for the words that have a cluster distance that falls in the bottom 50% of all cluster distances.

5 Corpus annotation and question answering.

In our experiments we allowed every word to be split into either one or two different meanings. Once a set of ambiguous words and their representation has been identified a new corpus is generated wherein the ambiguous words are annotated with their meaning. This is done by looping over the words in the corpus and again extracting the context of the words that are ambiguous. This context is then matched with either of the clusters that were found for that particular word. If the context of a word is closest to the representation of cluster 1, the word will be annotated with a '_1' and if the word context corresponds more with the second meaning of the word the word is annotated with '_2'. Then, the process described in (Mikolov et al., 2013a) is repeated to get the word-vector representation of the words in the annotated corpus. Now the question triplet'a is to b as c is to ...'.can be translated into many interpretations, namely 'a_0 is to b_0 as c_0 is to ...', 'a_1 is to b_0 as c_0 is to ...', etc. If all three words are ambiguous then there are 8 possible interpretations. Of course, not all interpretations are sensible. If the question is '*king* is to *queen* as *man* is to ...' then the interpretation of *queen* as a band is not sensible, but should be interpreted as 'queen as-in royalty'. To achieve this result we first ask the question 'which a and which b are most similar?' This is done by combining all senses of a and all senses of b and measuring their distance. It is assumed that royalty king and royalty queen are closer together than, for example, card-game king and the band queen. This way the question is reduced to two questions, where c is still ambiguous. Now both questions will be answered and an error $||\ e\ ||$ is determined for each answer, such that $b - a + c \equiv answer + e$. Finally we choose the answer with the smallest error.

6 Latent Semantic Analysis

XXX I think this might be better in the related work section, also because we don't have measurements for this method XXX
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).

7 Similarity Measures

XXX I don't think we should have a section on similarity measures. I think it would be good to mention it in the results section. XXX
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 dis-

tance 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:
$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 (y_i - \bar{y})^2}}$, 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.

8 Evaluation

We have evaluated the performance of COCONUT on the *enwik8*[1] dataset containing 60237 unique words, and has size 12577300. Initially, we decided not to disambiguated the top 75 most frequent words. XXX DESCRIBE QUESTIONS HERE XXX

8.1 Empirical Evaluation

For the empirical evaluation we have inspected the results of COCONUT. We have tested our data on the dataset we used to answer the analogy questions, but we have performed small tests on a hand-made dataset as well. The hand-made dataset contains different fragments of wikipedia articles on ambiguous topics, it includes apple (fruit, company), queen (band, monarch), jaguar (company, animal), eagles (band, animal), firm (law firm, firm grip), range (of numbers, farm fields) and more. For every ambiguous words we also took fragments from their superclasses. In addition to articles revolving around the ambiguous words, we have also added some random articles.

The results obtained on this small dataset were promising, for example the words from two different clusters (sorted based on relatedness) for the word 'apple' are:

1. also, fruit, june, announced, crisp, pie, crumble, inc, 9, is, apples, such, jelly, pomaceous, cake, 77, butter, processor, juice

2. iphone, wwdc, operating, develops, on, x, os, are, 4, sauce, desserts, nokia, remote, towards, offers, system, largest, worlds

Another example is 'jaguar' for which we find:

1. feline, fords, under, dropped, solitary, enjoys, waters, threatened, preferred, sustained, inland, 59, rainforest, swimming, across, ownership, largely, exceptionally, planned

2. models, sported, has, plated, traditionally, chrome, prominently, forming, famous, changed, hunts, grounds, associated, featured, americas, fishing

There is a fair amount of noise in the different clusters, but overall there is a reasonably clear distinction between the two different meanings of these words. On the *enwik8* dataset we find a lot more noise in the two clusters, including a lot of words that are relevant to both clusters. When inspecting the clusters the words with the highest relatedness to the disambiguated word, were most likely to be correct. However, the majority of the words that show little relatedness (even in they are in the top 50% of related words) do not describe the meaning as well. We provide two more examples of clusters formed on the *enwik8* dataset:

'santa'

1. claus, maria', monica, clara, tenerife, ana, san, croce, christmas

2. cruz, fe, barbara, catarina, grande, california, marta, mar, del

We observe a lot of noise in this example. However, the meaning of 'santa'-as-in Christmas is captured in meaning 1, whereas the meaning of 'santa'-as-in location is captured in meaning 2.

'belief'

1. god, faith, knowledge, justification, jesus, religion, absence, afterlife, resurrection

[1] http://cs.fit.edu/ mmahoney/compression/textdata.html

2. contrary, justified, atheism, systems, deities, beliefs, lack, freedom, feminism

Meaning 1 is more centered around 'belief'-as-in religion, whereas meaning 2 is more centered around a multitude of beliefs.

8.2 Quantitative Evaluation

9 Future work

XXX WORK OUT BULLET POINTS XXX

- would also be possible to use a soft-max or gaussian measure to weigh each word based on its distance, but we have weighted each of the 10 words in the 5 word window equally.
- multiple meanings
- better decision process for ambiguous or not
- cutting down on related words, we observe many not so related words
- capitalization can be a indication of a brand name

By inspecting the clusters, we found a lot of words that we would not consider ambiguous might occur in many different contexts An example of a word that we found has many different meanings is 'red'. This word had so many different meanings, that two word senses were not enough, we provide a limited breakdown of the words that co-occur with 'red':

- Boston Red Sox - sports
- Red Sea, Red Square - places
- Communism, love - concepts
- Relief, Red Cross - non-profits / brands
- ...

10 Conclusion

References

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013a. Linguistic regularities in continuous space word representations. Proceedings of NAACL-HLT, 746–751

Tomas Mikolov, Quoc V. Le and Ilya Sutskever. 2013b. Exploiting Similarities among Languages for Machine Translation. arXiv preprint arXiv:1309.4168,

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