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 10407936

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 of 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 analogy is '*seed* is to *apple* as *window* is to *house*' then the word apple refers to a fruit. Another example of an analogy is '*apple* is to *computer* as *porsche* is to *cars*' where the company apple is implied. It is clear that there are words that have two very distinct meanings. By representing words with just one vector (a continuous space representation), as done in (Mikolov et al., 2013a), it is not possible to differentiate between the two different meanings of a word. We propose to disambiguate between the various senses of a word in order to obtain multiple continuous space word representations for one word. We will annotate a training corpus to give ambiguous words two different representations. Then we reproduce the steps described in (Mikolov et al., 2013a) to obtain a continuous space word representation for the different meanings of a 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.

To obtain the continuous space word representations a recurrent neural network language model is used. XXX The RNN is trained using back-propagation 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.- MAYBE JUST GIVE THE IDEA OF RNNs, INSTEAD OF TELLING THIS. XXX

To answer the question 'a is to b as c is to ...', we use the word vectors x_a, x_b, x_c to determine the word which is assumed to be the best answer to a question. Following the example from the introduction, *queen* $-$ *king* $\approx$ *woman* $-$ *man*, we find that the answer to the question must be $y = x_b - x_a + x_c$. Since it is not likely that y computed in this matter actually exist, we find the answer to the question by maximizing the cosine similarity between y and the words in the corpus. We can also use this method in order to define a similarity between word pairs a:b and c:d. In this case we can answer less obvious analogy questions, we might for example want to find the relationship of competitors as denoted by the word pair *apple*:*dell*. When we are given two new word pairs *china*:*nike* and *ford*:*mercedes* we can rank them by computing the relational similarity: $cos(x_b - x_a + x_c, x_d)$.

Linguistic regularities can be used in many different unsupervised language learning applications. An example 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 observed 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 do not have a 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 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 assumption (1) we talk about 'words that accompany a word'. In COCONUT, words that accompany a word are co-occurring words. 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, the word we want to disambiguate. 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 contains the frequencies indicating how many times two words co-occur. We obtain the global co-occurrence vector for every word in the corpus in a similar way to how we obtaining the local co-occurrence vector as described in section 3. The only difference being that instead of maintaining all local co-occurrence vectors for a given word, we accumulate all local co-occurrence vectors in one global co-occurrence vector. This significantly reduces memory requirements which makes the COCONUT algorithm much more usable.

After obtaining the global co-occurrence vectors for every word in the corpus, we convert the absolute frequencies in the global co-occurrence vector to a relatedness score. 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

Let the set of co-occurrence vectors from the words in C, be called 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 disambiguated 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. We annotate a word with the index of the cluster that best describes its meaning. 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. We have now reduced the number of questions to only 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 Evaluation

We have evaluated the performance of COCONUT on the *enwik8*[1] dataset containing a total of 12577300 words and 60237 unique words.

We performed an empirical evaluating, in which we inspect the two meanings of a word. We also did a quantitative evaluation by using the continuous space word representations to answer a number of analogy questions. Different sets of analogy questions are available. For example the 8000 analogy questions used in (Mikolov et al., 2013a) contain adjectives (big, bigger, rough, rougher), noun (car, cars, apple, apples) and verb (avoid, avoids, wait, waits) questions. There are also many other analogy questions such as man-woman relations ships, country-currency relationships and more.

6.1 Similarity Measures

In (Mikolov et al., 2013a) the cosine similarity measure is used to find y in $y = b - a + c$. We have

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

Similarity Metric	Accuracy
Cosine	16.20
Euclidean	16.17
Manhattan	16.49

Figure 1: Results for answering 8000 syntactic analogy questions using the 80-dimensional word projections from (Mikolov et al., 2013a)

tried some other similarity measures or distances to see whether these would improve the accuracy on the set of 800 analogy questions proposed by the authors. 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. We report the accuracy for the cosine similarity as well as the euclidean distance and the manhattan distance, using the 80-dimensional word projections from (Mikolov et al., 2013a) that can be found online[2]. Table 6.1 shows the results for the different similarity measures. Although the manhattan distance slightly outperformed the cosine similarity we have chosen to use cosine similarity because it is not only the most intuitive similarity measure for this problem, but the computation can be performed very efficiently.

6.2 Empirical Evaluation

For the empirical evaluation we have inspected the results of COCONUT on a small hand-made dataset. 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 word 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 we noticed that 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 (XXXXX even if 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

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.

[2] http://rnnlm.org

6.3 Quantitative Evaluation

7 Future work

7.1 Co-occurrence vectors

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.

7.2 Detecting ambiguousness

One of the most challenging problems in word-sense disambiguation is deciding whether a word is ambiguous or not. At this point we discard all disambiguations that have a cluster distance that fall in the bottom 50%. In order to gain more insight in the distribution of these cluster distance we have made some plots. From these plots we see that most of the words are concentrated around a cluster distance of approximately 0.9. We do find that there are many words that have a cluster distance below 0.9 as well, in total this were X unique words, that together make up X$\%$ of the corpus. In future work, we might want to focus on finding a better way of detecting whether a word is ambiguous or not.

7.3 Detecting number of meanings

In our work we only split the words, i.e. we assume that a word can have at most two meanings. This is not a very solid assumption, we find many difference meanings including 1) financial institute 2) land alongside a river 3) cushion of a pool table. While inspecting the clusters, we found a lot of words that we would not consider ambiguous might occur in many different contexts, slightly changing the meaning of a word. 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
- ...

We believe that splitting the word vector for 'red' should still boost the performance when answering analogy questions. The analogy question '*red* is to *sox* as *blue* is to ...' might be difficult to answer when all different meanings of the word 'red' are encoded into one word representation. Once we are able to disambiguate the word 'red' we find a word representation that is focused at sports and not so much at communism. (the answer to '*red* is to *sox* as *blue* is to ...' is 'jays' as in Toronto Blue Jays, another question we could ask is '*red* is to *boston* as *blue* is to ...') In future work we should not restrict ourselves to a fixed number of meanings of one word, but investigate ways to determine how many meanings a word actually has.

8 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