COCONUT PALMs and Hand Palms: Improving Similarity Ratings by Word Sense Disambiguation

Anouk Visser
anouk.visser@me.com

Rémi de Zoeten
remi.de.z@gmail.com

Abstract

1 Introduction

When presented with two words without context it can be hard to assign a similarity to the two words. For example the word pair 'hit - single' could be interpreted as very similar because they are both related to music. However, hit can also be used in the context of hitting a ball, while single might be used to indicate someone's relationship status. To assigning similarity scores to word pairs the context in which words occur can be used to disambiguate the word properly. Word vector representations (Mikolov et al., 2013a) can be used to calculate the cosine similarity between two words, but these word vectors do not take context into account. In addition to this, these word vectors do not capture all meanings of a word specifically, as a word is represented only by a single vector.

In this report we research whether or not word sense disambiguation can help to improve similarity ratings on word pairs. We present three methods for word sense discrimination and disambiguation. COCONUT is a method for word sense discrimination that clusters the relatedness-vectors of the words co-occurring with the word we want to disambiguate. Agglomerative clustering is another clustering method that clusters word vectors based their cosine similarity in order to find clusters that represent different meanings that a word or a context might have. Finally, we present PALM which is a word sense discrimination method that produces SVMs which can be used for word sense disambiguation.

We evaluate our methods on a dataset constructed by (Huang et al., 2012) which consists out of 2003 word pairs in context that have been assigned a similarity score by human annotators.

2 Related Work

The task of word sense disambiguation is to assign the correct sense to an ambiguous word, whereas word sense discrimination is the task of finding the different senses a word might have (Schütze, 1998). The importance of using co-occurrences for determining the correct sense of a word is already emphasized in (Guthrie et al., 1991) in which the authors propose a method for word sense disambiguation using co-occurrences. More specifically, the authors propose a simple score expressing the relatedness between two words:

$$r(x, y) = \frac{f_{xy}}{f_x + f_y - f_{xy}} \tag{1}$$

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. In the past years a lot of different methods for word sense disambiguation have been proposed that can be classified as supervised, knowledge-based or unsupervised methods (Navigli, 2009). Unsupervised WSD mainly focuses on word sense discrimination. The majority of unsupervised word sense discrimination methods use clustering on, for example, context vectors or relatedness vectors. An example of an unsupervised method that uses clustering on both first order context vectors and second order context vectors can be found in (Purandare and Pedersen, 2004). Another example of unsupervised word sense discrimination is given in (Dinu and Lapata, 2010) where the authors use the intuition that the meaning of a word can be represented as a distribution over a set of latent senses.

Recently (Mikolov et al., 2013a) have released tools for efficiently computing word vectors that capture syntactic and semantic information. The distance between these vectors can be used for identifying linguistic regularities (Mikolov et al., 2013b) and a number of other applications such as word sense discrimination through clustering

these word vectors. However, word vector representations suffer from the problem that words may have a number of different meanings that cannot be captured in a single representation. (Reisinger and Mooney, 2010) propose a solution to this problem by representing a word's meaning by a set of sense specific word vectors which are discovered by clustering the contexts in which the word appears. For every context cluster, the authors compute an average vector that can be used to determine the similarity between two words (either in context or isolated). A drawback of this method is that it is required to know the number of senses in advance. (Huang et al., 2012) build upon this work by introducing a new neural network architecture that learns word vectors by also incorporating the global context of a word and can lean multiple vectors for a single word. In addition to this, they present a new dataset of pairs of words in contexts annotated with similarity judgements by human annotators.

3 Training data

For model training data we used the *enwiki8* dataset[1] corpus. For our purposes we filtered the corpus to only keep the words in the following Part-Of-Speech categories: Nouns, Verbs, Adjectives and Adverbs as is common in word sense disambiguation and discrimination systems (Navigli, 2009). Furthermore, we lemmatized all words so that e.g. *computer*, *computers* and *computing* are all projected onto the token *comput*. The result of this lemmatization is that the vocabulary shrinks and this means there are fewer variables.

4 COCONUT

The COCONUT method is a method for word sense discrimination and is based on two assumptions:

1. the meaning of a word is highly dependent on the words it co-occurs with

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

Let C be the set of words that co-occur with W, the word we want to disambiguate. COCONUT first constructs a global relatedness matrix containing relatedness vectors for every word in the corpus and the words that co-occur with it according to equation 1. k-means clustering is applied to the relatedness vectors of the words in C to provide k bags of words describing the k different senses of W. Some examples of the resulting clusters can be found in figure 1. A significant disadvantage of COCONUT is that the number of senses for a word has to be known in advance.

[1] http://mattmahoney.net/dc/textdata.html

5 Agglomerative Clustering

One way to cluster data is agglomerative clustering which has been shown to produce good results in comparison with other clustering techniques (Purandare and Pedersen, 2004). Agglomerative clustering is an iterative bottom-up approach to clustering. Initially each data point forms its own cluster and in each iteration the two data points that are closest to each other are merged until there is only a given number of clusters or the inter-cluster distances are larger than a predefined threshold. We performed agglomerative clustering on the 80 dimensional word vector representations that we extracted from our training data. We clustered the words into 500 clusters. Of these 500 clusters 351 were single word clusters. The distribution over the number of words in each cluster is skewed and clusters with one or just a few words in them are not informative. Therefore we removed the clusters that had less than 10 words in them, which resulted in 41 clusters with a distribution over the cluster size shown in figure 1.

5.1 Comparing word context with clusters

We will define two methods for using the word clusters to define a word relatedness score. In both methods we compare the context of each word with each of the clusters. We define a probability $P(cluster|context)$ for each cluster, and produce a normalized vector V_p of probabilities for each context. The distance between two words is then defined by the cosine similarity between these two vectors.

5.1.1 Cluster - context intersection

In the first instance we determine the likelihood of a context being represented by a cluster by counting how many words are in common with the cluster:

$$P(cluster|context) = \frac{|\{cluster \cap context\}|}{|cluster|} \quad (2)$$

bat	course	bank
batter, superfamilies, inning, ye, ball, batman, cave, ruth, pitch, slug, hitter, base, plate, flies, mammal	hole, student, action, educate, online, learn, studies, disc, teach, meal, hungarian, university, employee, lecture, play	reserve, imf, central, account, money, finance, european, deposit, sweden, invest, intern, cccc, palestinian, note, economic, tower
average, hit, cricket, baseball, out, funnel, casey, score, rabies, runner, myotis, ab, statist, league, pollin	golf, caddie, require, taught, offer, historic, distance, college, typic, qualify, event, year, decide, entire, business	gaza, strip, monetary, financial, river, fund, feder, currency, settlement, england, loan, sector, israel, jordan, isra

Table 1: Table showing the 15 most related words in the two sense clusters (clustered with $k = 2$) of the words *bat*, *bank* and *course*. For bat we observe a lot of noise, however the majority of words that describe bat-as-in animal are in the first cluster (superfamilies, batman, cave, flies, mammal). The first sense of course is focussed towards learning (student, educate, online) as well as meals, the second sense also contains the words golf, caddie and distance which refer to course-as-in golf. Finally, for bank we find that the second cluster contains word referring to river (river, settlement, strip) as well as words referring to the middle east (gaza, israel, jordan). The first cluster is more focussed towards bank as a financial institution.

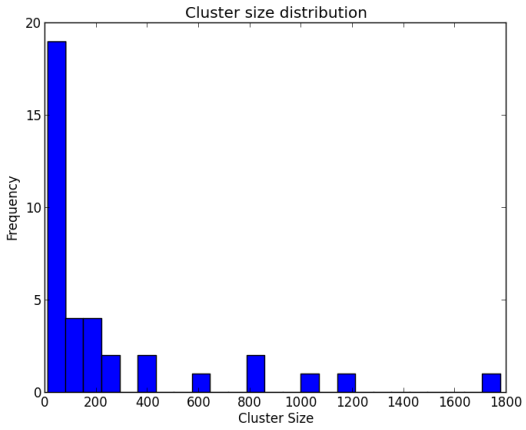

Figure 1: Distribution over cluster sizes for agglomerative clustering. As can be seen there are mainly small clusters with up to 200 words, but there exist also larger clusters.

By applying equation 2 to all clusters for a given context and normalizing over the different probabilities we get a probability vector V_p that defines a mixture of the context over the different clusters. The similarity score for two words is defined by the cosine similarity between the two probability vectors of the two contexts. The similarity that we find using this method can be smoothed with the similarity of the single word vectors of the words that we are discriminating.

5.1.2 Expanded Cluster - context similarity

In this method we do not directly compare the words in a cluster with the context, but replace the clusters with the average of the co-occurrence vectors of the words in the cluster. Co-occurrence vectors are extracted from the corpus by finding every instance of a given word and observing the 5 words that occur before and after that word. The co-occurrence vector is a normalized sparse vector that defines the probability for a word co-occurring with a given other word. This extended cluster representation can be compared with a context of a word by cosine similarity of the two vectors. This is how two contexts are compared with every cluster and again cosine similarity can be used to define how related two contexts are.

6 PALM

PALM (Probabilistic Agglomeratively-clustered Latent Meanings) is a method for word sense discrimination that simultaneously trains an SVM for

every word that can be used for word sense disambiguation. The SVM is able to disambiguate a word by predicting a label best describing the sense of a word when given the probability distribution from the word's expanded context over the agglomeratively-clustered latent meanings (that were obtained using the clustering method described in section 5). The predicted labels can for example be used to relabel a corpus before training a recurrent neural network in order to obtain multiple vector representations for one word.

In this section we describe the PALM method in detail, figure 2 provides an overview of the algorithm.

6.1 Choosing the label

Let W be the word for which we want to train the SVM. PALM starts by extracting all contexts from a corpus that W appears in. We define 'context' as all words within a window around W (in our experiments we looked five words back and five words ahead). Our aim is to assign a label to each of these contexts that describes the sense of the words best, the collection of labels then represent the different senses a word can have. As we have seen in (Jurgens, 2014) underspecified contexts are often observed. In line with our assumption for the COCONUT baseline (i.e. 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) we expand the context by adding the n (in our experiments we set $n = 5$) most related words to every word in the context (except for W itself). Finally, a word w from the expanded context is selected as a label so that:

$$label = \arg\max_{w} r(W, w) + \textit{sim}(W, w) \quad (3)$$

where $r(W, w)$ is the relatedness score from equation 1 and $\textit{sim}(W, w)$ denotes the cosine similarity between W and w.

6.2 Probability distribution over agglomeratively-clustered latent meanings

Let P be an N-dimensional vector describing the probability distribution of a word's context over the clustered meanings. We compute an element p_i in P as follows:

$$p_i = \frac{1}{N_c} \sum_{w \in C} sim(w, m_i) \quad (4)$$

bat	course	bank
bird	need	invest
slugger	education	west
inning	even	gaza
shark	golf	foreign
ye	teach	central
	learn	trade
	have	fund
		finland

Table 2: Labels for *bat*, *course* and *bank* after label reduction. The labels are sorted in order of appearance where the top labels are the most frequent and the bottom labels least frequent.

where N_c is the number of contexts, m_i is the vector representation of the i^{th} cluster (i.e. latent meaning) and C is the collection of all words in the expanded context. Although the values denote the accumulated similarity of the words from the expanded context to the meanings, we can interpret the vector as a probability distribution of a word's context over clustered meanings (when the context and a meaning are very similar, it is very likely that the context imposes this meaning).

6.3 Label reduction

The last step before training the SVM consists of reducing the number of labels. By expanding the context from which the label is selected, we increase the likelihood of selecting the same label multiple times, but many superfluous labels will remain. We apply a modified version of agglomerative clustering on the labels in order to reduce the amount and assure that only 'clusters' are formed that represent the same meaning. We have implemented the modified agglomerative clustering method so that it:

- favors labels that were observed most frequently
- does not require setting the number of remaining clusters in advance

To favor labels that were observed most frequently we split the labels into two halves: the upper half (containing labels that were seen most frequently) and the lower half (containing labels that were seen least frequently). We restrict the clustering method to merging two labels of which one of them is in the lower half (w_l) and one in the upper half (w_u), so that labels that occurred frequently

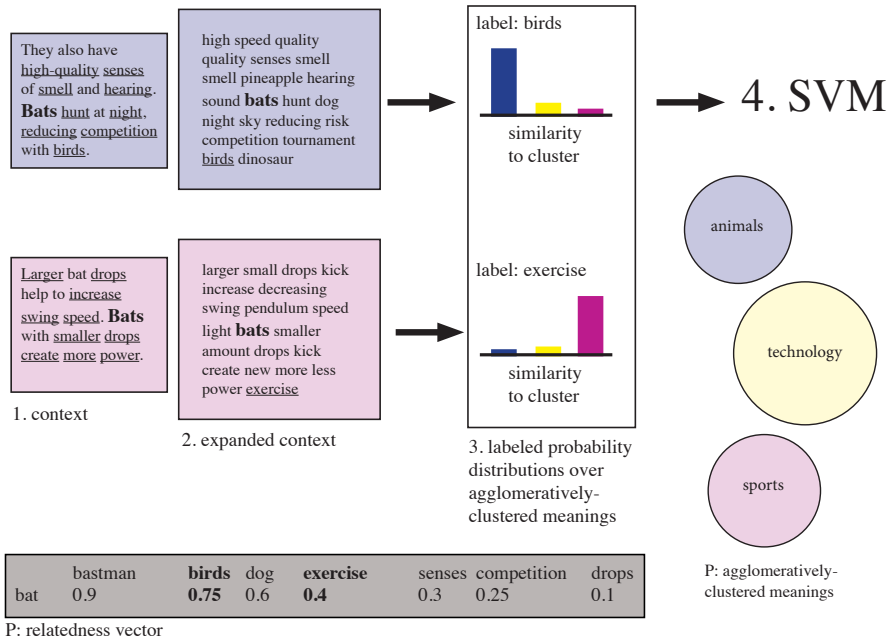

Figure 2: The five steps of the PALM algorithm. P: preprocessing, PALM requires agglomeratively-clusterd latent meanings and relatedness vectors for all words in the corpus. 1. Extract every context word W appears in. 2. Expand the context. 3. Choose a label from the expanded context and construct the vector representing the probability distribution from the word's expanded context over the agglomeratively-clustered latent meanings for every context. 4. Train an SVM on these probability distribution vectors using the selected labels.

are less likely to be merged into another label. w_l and w_u are selected so that:

$$\arg\max_{w_l, w_u} \textit{sim}(w_l, w_u)$$

This process continues until $\textit{sim}(w_l, w_u)$ is below a certain threshold (in our experiments the threshold was 0.5). By using the similarity as a threshold the final number of clusters depends only on the observed data. Table 2 shows the remaining labels for the words *bat*, *course* and *bank*. The reordered labelled data can then be used to train an SVM.

7 Experiments

We evaluate the performance of our WSD methods on the dataset constructed by (Huang et al., 2012). The dataset consist of 2003 word pairs and their context. The goal of the task is to assign a similarity measure to all word pairs. 241 word pairs consist of the same word leaving a total of 1712 unique words. Ten human judges assigned similarity scores to all word pairs. Table 3 contains four example word pairs with their context and the scores assigned by two methods and the human annotators. We compute the

WSD Method	$\rho \times 100$
Single Word Vector (SWV)	60.1
COCONUT	38.2
Agglomerative	19.5
Agglomerative + SWV	60.4
Agglomerative Extended	54.5
PALM	49.9
Joint PALM	57.3

Table 4: The results

Spearman correlation between the method's similarity ratings and the average rating of the human annotators. We compare our methods to the Single Word Vector (SWV) baseline that uses only a single word vector for every word and computes the similarity as the cosine similarity between the two words without taking the context into account.

As mentioned in section 4 COCONUT is a method for word sense discrimination. We disambiguated all 1712 words into two different senses ($k = 2$), a sense is represented as a collection of words that indicate this sense. As a representation

Word 1	Word 2	PALM label1 / label2 / score	Human	SWV
In northern New **Mexico**, the local ”black on white” tradition , the Rio Grande white wares, continued well after 1300 AD .	Women’s basketball was added to the Olympics in 1976, which were held in Montreal, Canada with teams such as the Soviet Union, **Brazil** and Australia rivaling the American squads.	canada / cup / 0.58	0.44	0.83
It has an aromatic, warm and slightly **bitter** taste.	AK - a very common beer name in the 1800s - was often referred to as a ”mild **bitter** beer” interpreting ”mild” as ”unaged”.	taste / war / 0.49	0.75	1.0
Shockley took the lion’s share of the **credit** in public for the invention of transistor, which led to a deterioration of Bardeen’s relationship with Shockley .	Payment in kuna, all major **credit** cards and euros are accepted at all toll gates.	invent / pay / 0.48	0.31	1.0
Located downtown along the east **bank** of the Des Moines River, the plaza is available for parties, social events, movies, concerts, and summer sand volleyball during the warmer months of the year.	This is the basis of all **money** laundering, a track record of depositing clean money before slipping through dirty money.	west / pay / 0.57	0.25	0.75

Table 3: Four example word pairs from the dataset. For these examples we provide the labels for the two words given by the PALM disambiguation method, including the similarity scores assigned by PALM, SWV and the human annotators.

for the words we choose to use the average word vector of all words that belong to the most appropriate sense of the word given the context, which we define as:

$$\arg\max_{sense} |\{sense \cap context\}| \quad (5)$$

where *sense* is the set of words in one of the two senses of the disambiguated word and *context* is the set of words in the context.

To compute the score for PALM we applied the word sense discrimination phase on the *enwiki8* corpus. PALM was able to identify different senses for 1222 of the 1712 unique words of the word pairs, resulting in 1222 SVMs (we found that an average of 4.45 senses are assigned to a word with a standard deviation of 3.15). We then relabeled all occurrences of the 1222 disambiguated words in the corpus by appending the label predicted by the SVM for the words. Finally, we used *word2vec* by (Mikolov et al., 2013a) to obtain 80-dimensional word vectors for all words in the relabeled corpus. For a word pair in the task, we can again obtain the label and use this to find the correct word vector. These word vectors are used to compute the cosine similarity between the two words. Joint PALM is a variation on PALM that computes the similarity between the average word vectors of the single word vector and the word vector selected by PALM.

The results of the WSD methods can be found in table 4.

7.1 Analysis of results

When comparing our methods to the baseline, only Agglomerative + SWV improves it slightly. COCONUT was not designed for word sense disambiguation and a lot of different variants can be implemented that differ in the way they represent a word. As we observed in table 1 the clusters contain a lot of noise that decrease the quality of the word representation when it is computed as the average of the words in the most appropriate sense. Also, COCONUT splits every word in a

fixed number of senses, which may not lead to an improvement when the word is not ambiguous at all.

Table 3 contains four examples of word pairs and the ratings provided by the human annotators, PALM and SWV. For the first word pair (Mexico - Brazil) SWV assigns a much higher similarity score than PALM. When looking at the labels found for countries we often find that they are related to the different contexts that a country's name can occur in (e.g. sport events, other countries, export, foreign, military, ...). We find that PALM is very sensitive to the different meanings of countries and cities, which does not correlate strongly with the way human annotators assign similarity scores when presented word pairs containing a country or city. The second example shows that when PALM does not properly disambiguates the word it affects the similarity scores negatively. However, when PALM does assign the correct label to the words as can be seen in word pairs credit - credit and bank - money, it can be more successful in assigning a similarity score than SWV.

8 Conclusion

We designed, implemented and evaluated various methods for context based word sense disambiguation and compared their performance to the COCONUT and Single Word Vector baselines. Our methods are based on word clusters and show different ways to use these for word sense disambiguation. We showed that WSD based on context can perform comparable to Single Word Vector discrimination. We were unable to show that WSD significantly improves the spearman correlation on the task of similarity ratings.

References

Georgiana Dinu and Mirella Lapata. 2010. Measuring distributional similarity in context. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1162–1172. Association for Computational Linguistics.

Joe A Guthrie, Louise Guthrie, Yorick Wilks, and Homa Aidinejad. 1991. Subject-dependent co-occurrence and word sense disambiguation. In *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, pages 146–152. Association for Computational Linguistics.

Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics.

David Jurgens. 2014. An analysis of ambiguity in word sense annotations. In *Proceedings of LREC*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751. Citeseer.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10.

Amruta Purandare and Ted Pedersen. 2004. Word sense discrimination by clustering contexts in vector and similarity spaces. In *Proceedings of the Conference on Computational Natural Language Learning*, volume 72. Boston.

Joseph Reisinger and Raymond J Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117. Association for Computational Linguistics.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational linguistics*, 24(1):97–123.