Learning Thesaurus Relations from Distributional Features

Rosa Tsegaye Aga, Christian Wartena, Lucas Drumond, Lars Schmidt-Thieme
Hochschule Hannover, Universitat Hildesheim
rosa-tsegaye.aga@hs-hannover.de, christian.wartena@hs-hannover.de, ldrumond@ismll.de,schmidt-thieme@ismll.info

Abstract

In distributional semantics words are represented by aggregated context features. The similarity of words can be computed by comparing their feature vectors. Thus, we can predict whether two words are synonymous or similar with respect to some other semantic relation. We will show on six different datasets of pairs of similar and non-similar words that a supervised learning algorithm on feature vectors representing pairs of words outperforms cosine similarity between vectors representing single words. We compared different methods to construct a feature vector representing a pair of words. We show that simple methods like pairwise addition or multiplication give better results than a recently proposed method that combines different types of features. The semantic relation we consider is relatedness of terms in thesauri for intellectual document classification. Thus our findings can directly be applied for the maintenance and extension of such thesauri. To the best of our knowledge this relation was not considered before in the field of distributional semantics.

Keywords: Distributional semantics, thesauri, context vectors, Supervised machine learning

1. Introduction

In the past decades, distributional similarity (DS) has been used to solve many different tasks related to the meaning of words. The main idea of DS is, that words that appear in similar contexts are likely to have a similar meaning. When context representations are built from large amounts of texts, DS can give good results for various tasks. However, DS does not directly correspond to any traditional semantic relation. Distributional similar words might be synonyms, antonyms, hypernyms etc.

In this paper we will consider a very basic and central task for computational lexical semantics: the decision whether two words are semantically related or not. A simple approach is to compute a similarity between the words and learn a threshold above which the words can be considered as being related. We show that a much stronger supervised approach, in which the similarity between the words itself is learned from examples, gives much better results. To do so we need to construct feature vectors representing pairs of words. Especially, we show that a simple combination of context vectors gives better results than using vectors combining different types of features.

Much research on DS has been done on very small datasets; For example, the TOEFL data, presumably the most often used dataset in this field, just has 80 questions. Because we need large datasets, we use data sampled from large thesauri, especially Eurovoc (Office for Official Publications of the European Communities, 1995) and STW (ZBW - Leibniz Information Centre for Economics, 2014). These data have two advantages: It is easy to sample large amounts of related (and non-related) words and the data are constructed independent of any specific semantic task.

Since thesauri are also used for automatic indexing and for full-text retrieval, it is important to know all possible terms that refer to a certain concept. Therefore, extension of thesauri with more labels for each concept is an important task in the maintenance of these vocabularies. Thus the results of this paper also have a direct practical application.

The rest of the paper is organized as follows. In section 2., we review related work. Section 3. explains the distributional feature construction and pairwise feature generation. We discuss the experiment data, supervised method and evaluation in Section 4.. In Section 5. and 6., we discuss the result and conclusion with future work respectively.

2. Related Work

Distributional similarity can be seen as a machine learning method, since we derive semantic representations from large amounts of data. Nevertheless, supervised methods have not been very popular in this field. Zhitomirsky and Dagan (2009) use a supervised method for feature selection. In (Bär et al., 2012) and (Wartena, 2013) ensemble models are used to combine different DS measures.

The approach of Turney (2014) also is based on supervised learning using a large number of different similarity features. The difference to the afore mentiones approaches is, that Turney focusses on similarity of phrases and the similarities that are used as features are not similarities for the pair themselves.

The work that is most closely related to ours is the research on metric learning for DS from (Shimizu et al., 2008) and (Hagiwara, 2008). Shimizu et al. (2008) used a learned Mahalanobis distance to rank pairs of synonyms and unrelated words. In order to make the learning computationally feasible they reduced the number of context features massively by selecting the most promising features. In the approach of Hagiwara (2008) feature selection is not necessary as he constructed features to represent each pair of words. Subsequently a Support Vector Machine (SVM) (Cristianini and Shawe-Taylor, 2000) is used to learn which pairs are pairs of synonyms and which pairs are not. We will follow a similar approach but we use different ways to construct the features.

One of the practical applications of DS that is often mentioned, is automatic updating and extension of a thesaurus with new terminology (Crouch, 1990; Curran and Moens, 2002; Turney and Pantel, 2010). Wartena and Brussee (2008) use DS to align free tags with categories from Wikipedia. However, we are not aware of any attempt to extend large existing thesauri with new terms or concepts

Table 1: Pair of words datasets size

Classes \ Datasets	SC53	Eurovoc-2	Eurovoc-1	Eurovoc-0	STW-1	STW-0
Positive	1 752	10 000	10 000	2 175	10 000	10 000
Negative	1 752	10 000	10 000	2 335	10 000	10 000

based on DS.

3. Methods

The task that we consider is deciding whether a pair of words is semantically related. We construct vectors of co-occurrence features to represent each word. However, instead of computing a similarity between two feature vectors we will construct a vector of features representing a pair of words. A linear Support Vector Machine (SVM) is trained on the features of the word pair to discriminate between related and unrelated pair of words. The SVM model decides whether two words are semantically related by considering the feature vector representing the pair of words. We built models on two types of vectors: 1. Vectors constructed by a simple operation on context vectors, 2. Vectors constructed by aggregating different types of features using Turney's *SuperSim* method (Turney, 2014). The two types of vector construction will be explained in section 3.2. .

We compared the results between the two models and with results obtained by using cosine similarity. For cosine similarity training is nothing more than finding an optimal value to split the examples.

3.1. Feature construction

In DS the meaning of a word is represented by a vector of context features. As context features co-occurrence data with other words in a large text corpus are used.

There are a number of choices that have to be made when building the context vectors for each word. In the following we will use the choices that turned out to yield the best results in a number of different tasks in recent studies by Bullinaria and Levy (2007; 2012) and Kiela and Clark (2014). First it has to be determined what words are used as context features, i.e. for what words co-occurrence statistics have to be computed. Generally, it is found that mid frequency words are most effective. After some preliminary experiments we found that including all words in the frequency range from $4 \cdot 10^3$ to $1 \cdot 10^6$ in the UkWaC Corpus is a good compromise between optimal results and acceptable storage and computing efforts. Therefore, context words which have frequency range from $4 \cdot 10^3$ to $1 \cdot 10^6$ in the UkWaC Corpus have considered to construct the context vector for each words. Then each word is now represented by a vector of 17 400 features. In other word, there are 17 400 different words that occur at least $4 \cdot 10^3$ times and at most $1 \cdot 10^6$ times in the corpus.

Next we have to determine the size of the window for co-occurrence. If the training corpus is large enough all studies show that smaller windows yield better results. We first remove all stop words and then use a window size of two words on the stopped text, respecting sentence boundaries. Syntactic relations are not used to determine the context of a word.

Finally, we use positive pointwise mutual information (PPMI) as a degree of co-occurrence, since it was shown to give better results than raw co-occurrence probabilities in a number of different studies. For a context words c and a (target) word t the PPMI is defined as

$$ppmi(c,t) = \max\left(\log \frac{p(c|t)}{p(c)}, 0\right). \tag{1}$$

3.2. Representation of word pairs

In order to decide whether two words are semantically related, Shimizu (2008) proposed learning a SVM model by taking the distributional features as an input, that were constructed by addition of the context vectors of both words. Turney (2014) proposes a method, called *SuperSim*, to represent pair of words, and identify related pairs. Turney represents a pair of words by aggregating four types of features, all based on frequencies in a large corpus. These are logarithm frequency values for each word of the pair, PPMI between the two words of the pairs, the similarities of two words in domain space, and the similarity of two words in function space Turney (2014).

To find the best representation for a pair of words, we compared the above three approaches. For the first approach, we moreover compared four different methods to combine the two co-occurrence feature vectors. These four proposed methods are:

- Addition; used before by Shimizu (2008) who obtained good results using this method.
- Subtraction
- Point-wise or Hadamard product (closely related to the cosine: if we normalize the length of the vectors, the cosine is the sum of the elements of their Hadamard product)
- Binary vector. The binary vector $\overrightarrow{v}_{bin}$ of two vectors $\overrightarrow{v}_1$ and $\overrightarrow{v}_2$ is defined by setting $\overrightarrow{v_{bin,i}} = 1$ if $\overrightarrow{v_{1,i}} > 0$ and $\overrightarrow{v_{2,i}} > 0$, otherwise $\overrightarrow{v_{bin,i}} = 0$.

For the addition, subtraction and point-wise, we also test the variants in which we first normalize the length of the vectors to the unit vector.

4. Experiment

In this section, we will describe the experimental setup and the data used.

4.1. Data description

For building the context vectors, we used two different corpora: UkWaC for English and DeWaC (Baroni et al., 2009) for German.

Table 2: Accuracy of *synonymous* and *non synonymous* word pair classifiers

Method		**Datasets**						
	Word Pair Feature(s)	**SC53**	**Eurovoc-2**	**Eurovoc-1**	**Eurovoc-0**	**STW-1**	**STW-0**	**Average**
Split	**Cosine**	0, 87	0,71	0,77	0,75	0,65	0,65	0,73
Linear SVM	**SuperSim**	0,87	0,77	0, 69	0,73			0,77
	Binary	0,96	0,99	0,85	0,71	0,80	0,63	0,82
	Addition	0,94	**1**	0,92	0,65	0,81	0,53	0,81
	Subtraction	0,99	**1**	0,94	0,73	0,81	0,55	0,84
	Multiplication	0,96	0,99	0,86	0,72	**0,82**	0,66	0,84
	Addition (unit vectors)	**0,99**	0,99	**0,95**	0,74	0,81	0,55	0,84
	Subtraction (unit vectors)	0,99	**1**	0,94	0,73	0,81	0,55	0,84
	Multiplication (unit vectors)	0,97	0,92	0,88	**0,84**	0,80	**0,71**	**0,85**

For our experiments we use data extracted from two large thesauri and from a data set introduced in (Bullinaria and Levy, 2007). This last data set contains 530 words which have been taken from 53 semantic categories (10 words for each category). We will refer to this data set as SC53. From this collection we draw a set of 1752 pairs of words belonging to the same category and 1752 randomly chosen pairs of words from two different categories.

All other data sets are extracted from thesauri. Since thesauri are organized hierarchically we can define similarity at different levels. We use both similarity at a very fine grained level, considering almost only synonyms as similar and at a broad level, considering all terms belonging to the same branch of the hierarchy as similar. However, since each thesaurus has some focus domain that is worked out very detailed and other areas that are modeled much more coarse grained, in the core of the thesaurus two terms denoting the same concept will be real synonyms, while in other areas quite different words can refer to the same concept. Keeping this in mind, we understand that an automatic approach will never be able to decide whether two terms denote the same concept without errors .

We have compiled three data sets from the Eurovoc Thesaurus (Office for Official Publications of the European Communities, 1995). Eurovoc is a multilingual thesaurus developed by the European Commissions Publications Office as a controlled vocabulary for the intellectual indexation of documents. The Eurovoc thesaurus is divided into 127 micro-thesauri. From each of these micro-thesauri we took the top-level concepts, 528 in total, as semantic categories. For each category we collected all narrower concepts and considered their preferred and alternative labels as terms for that category. We then removed all terms that belong to more than one category or that consist of more than two words. Finally, we removed all categories for which less than 10 terms were found. Now 190 categories with a total of 2386 terms are left. The largest category consists of 44 terms. From this data set we have constructed two set of pairs: the first set has 10 000 pairs of words belonging to the same category and the second set has 10 000 randomly chosen pairs from two different categories. We will refer to this set of pairs a Eurovoc-1, since the terms are equivalent by going up one level in the Eurovoc concept hierarchy. Furthermore we built a collection of pairs by selecting 10 000 pairs of words from the same data set where both words are taken from the same micro-thesaurus and 10 000 pairs taken from two different micro-thesauri. We call this set Eurovoc-2. Both for Eurovoc-1 and Eurovoc-2 we only selected terms consisting of one word and we moreover required that word occurs at least once in the UKWaC.

Finally, we sampled pairs of words from the original Eurovoc thesaurus by taking preferred and alternative labels for the same concept as synonymous terms, and pairs that are used as labels for different concepts as non-synonym pairs. For the negative examples we want to have an equal distribution of easy and difficult pairs. Thus we took 20% pairs with words from concepts with a distance of 1 step, using any specified thesaurus relation. Further 20% were taken from concepts with a distance of 2 steps, and so on. For the last 20%, pairs of concepts with a distance of at least 5 steps were used. In all cases it was ensured that no shorter path exists. For this set, we have 2175 synonymous and 2335 non-synonymous words; This set of pairs we will refer to as Eurovoc-0.

The German experiment data are derived from German notations from the German thesaurus on business and economics *Standard-Thesaurus Wirtschaft* (STW). The STW is divided into 7 sub-thesauri. Each part consists of a hierarchy of notations and descriptors. Descriptors have preferred and eventually non-preferred labels and in some cases narrower descriptors. We took all terms (i.e. labels from descriptors) from 6 sub-thesauri (leaving out the sub-thesaurus with general terms) that belong to only one notation and consist of at most 2 words. Subsequently, we removed all words belonging to a notation with less than 5 terms in our sample. This gives us 419 classes (one class for a notation) with a total of 11 599 terms. There are 5 classes with over 100 terms. The largest class has 233 terms. From this set we randomly selected a set of pairs of words, 10 000 from the same class and 10 000 from different classes. We restricted the selection of words to words occurring at least once in DeWaC, but we included multi-words. We call this set STW-1.

Finally, we selected pairs of words where the terms are labeled for the same or different descriptors. We controlled the distribution of negative pairs in the same way the Eurovoc-0 data. This data set has 10 000 positive and 10 000 negative pairs and is referred to as STW-0.

The pair of words data size is shown in Table 1.

4.2. Supervised Similarity Learning

In this section, we will see how we used the supervised method to predict the pair of words by taking the distributional features as an input.

We used linear SVM from the liblinear package to learn a model and classify the word pairs represented by one feature vector. Liblinear is very efficient and fast for training large-scale problems (Fan et al., 2008). The hyperparameters of the models have been tuned using grid search from LIBSVM. To find the best C parameter value, we tested the numbers in between 0 and 20 in step 0.05.

4.3. Experiment Setup

We evaluated the models presented in Section 3.2. on distinguishing pairs of related words from arbitrary pairs using ten-fold cross validation.

For all experiments, we used ten cross validation. We considered 10% for the test and 90% for the train set, and generated ten train and ten test datasets. For each methods the same split was used.

5. Results

We can consider the results obtained by using cosine similarity as a baseline, since the cosine is usually considered as the best similarity measure for classification of context vectors (Bullinaria and Levy, 2007; Bullinaria and Levy, 2012; Kiela and Clark, 2014). Another possible baseline is Turney' SuperSim method (Turney, 2014) that gave good results for phrase similarity in a number of experiments.

In total we have done 48 classification experiments on both English and German pairs, and 3 more classification experiments on different types of features (Turny's approach) only on English pairs. The average accuracy results from ten-fold cross-validation are given in Table 2. The results confirm the outcome of the experiment from (Hagiwara, 2008) on one data set: training an SVM on one type of features representing the pairs yields better results than using cosine similarity and different types of features classification. We find that a SVM learned on the pairwise features on one type for all 6 data sets gives better results than the cosine similarity and different types of features classification. However, feature addition, as used by Hagiwara, seems not always to be the best choice and only gives best results when the feature vectors are normalized before addition.

6. Conclusions and Future Work

Our experiments suggest that a model learned on one type of feature for a specific task can give better results than a general similarity measure and different types of features. In our experiment, we have shown that multiplication and addition are the best methods to construct pairwise features, and learn the pairs relation on SVM. Furthermore, we see that the vector length allays should be normalized before vectors are combined by one of the four methods to represent a word pair.

For further work, we would like to extend the method to classification into more than two classes, and to investigate more specific relations between words.

7. Bibliographical References

Bär, D., Biemann, C., Gurevych, I., and Zesch, T. (2012). UKP: Computing Semantic Textual Similarity by Combining Multiple Content Similarity Measures. In *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval-2012)*, pages 435–440.

Baroni, M., Bernardini, S., Ferraresi, A., and Zanchetta, E. (2009). The WaCky Wide Web: A Collection of Very Large Linguistically Processed Web-Crawled Corpora. *Language Resources and Evaluation 43 (3): 209-226*, 43(3):209–226.

Bullinaria, J. A. and Levy, J. P. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behaviour Research Methods*, 39(3):510–526.

Bullinaria, J. A. and Levy, J. P. (2012). Extracting Semantic Representations from Word Co-occurrence Statistics: Stop-lists, Stemming and SVD. *Behaviour Research Methods*, 44(3):890–907.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and other kernel-base learning methods*. Cambridge University Press.

Crouch, C. J. (1990). An approach to the automatic construction of global thesauri. *Information Processing & Management*, 5:629–640.

Curran, J. R. and Moens, M. (2002). Improvements in Automatic Thesaurus Extraction. In *Unsupervised Lexical Acquisition: Proceedings of the Workshop of the ACL Special Interest Group on the Lexicon (SIGLAX)*, pages 59–66.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, June.

Hagiwara, M. (2008). A Supervised Learning Approach to Automatic Synonym Identification Based on Distributional Features for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA, Student Research Workshop. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 1–6.

Kiela, D. and Clark, S. (2014). A Systematic Study of Semantic Vector Space Model Parameters. In *2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*.

Office for Official Publications of the European Communities. (1995). Thesaurus eurovoc - volume 2: Subject-oriented version.

Shimizu, N., Hagiwara, M., Ogawa, Y., Toyama, K., and Nakagawa, H. (2008). Metric learning for synonym acquisition. In *COLING '08 Proceedings of the 22nd International Conference on Computational Linguistics*, pages 793–800.

Turney, P. D. and Pantel, P. (2010). From Frequency to

Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Turney, P. (2014). Distributional semantics beyond words: Supervised learning of analogy and paraphrase. *Transactions of the Association for Computational Linguistics*, 1:353–366.

Wartena, C. and Brussee, R. (2008). Instance-Based Mapping between Thesauri and Folksonomies. In Amit Sheth, editor, *The semantic web - ISWC 2008*, volume 5318 of *Lecture Notes in Computer Science SL 3, Information Systems and Application, incl. Internet/Web and HCI*, pages 356–370. SpringerLink [host], Berlin.

Wartena, C. (2013). HsH: Estimating semantic similarity of words and short phrases with frequency normalized distance measures. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012), Atlanta, Georgia, USA*.

ZBW - Leibniz Information Centre for Economics. (2014). STW Thesaurus for Economics.

Zhitomirsky-Geffet, M. and Dagan, I. (2009). Bootstrapping Distributional Feature Vector Quality. *Computational Linguistics*, 25(3):435–461.