HsH: Estimating Semantic Similarity of Words and Short Phrases with Frequency Normalized Distance Measures

Christian Wartena
Hochschule Hannover – University of Applied Sciences and Arts
Department of Information and Communication
Expo Plaza 12, 30539 Hannover, Germany
Christian.Wartena@hs-hannover.de

Abstract

This paper describes the approach of the Hochschule Hannover to the SemEval 2013 Task *Evaluating Phrasal Semantics*. In order to compare a single word with a two word phrase we compute various distributional similarities, among which a new similarity measure, based on Jensen-Shannon Divergence with a correction for frequency effects. The classification is done by a support vector machine that uses all similarities as features. The approach turned out to be the most successful one in the task.

1 Introduction

The task *Evaluating Phrasal Semantics* of the 2013 International Workshop on Semantic Evaluation (Manandhar and Yuret, 2013) consists of two subtasks. For the first subtask a list of pairs consisting of a single word and a two word phrase are given. For the English task a labeled list of 11,722 pairs was provided for training and a test set with 3,906 unlabeled examples. For German the training set contains 2,202 and the test set 732 pairs. The system should be able to tell whether the two word phrase is a definition of the single word or not. This task is somewhat different from the usual perspective of finding synonyms, since definitions are usually more general than the words they define.

In distributional semantics words are represented by context vectors and similarities of these context vectors are assumed to reflect similarities of the words they represent. We compute context vectors for all words using the lemmatized version of the Wacky Corpora for English (UKWaC, approximately 2,2 billion words) and German (DeWaC, 1,7 billion words) (Baroni et al., 2009). For the phrases we compute the context vectors as well directly on the base of occurrences of that phrase, as well as by construction from the context vectors of the two components. For the similarities between the vectors we use Jensen-Shannon divergence (JSD) and cosine similarity. Since the JSD is extremely dependent on the number of occurrences of the words, we define a new similarity measure that corrects for this dependency. Since none of the measures gives satisfactory results, we use all measures to train a support vector machine that classifies the pairs.

The remainder of this paper is organized as follows. We start with an overview of related work. In section 3 we discuss the dependence of JSD on word frequency and introduce a new similarity measure. Section 4 then describes the system. The results are given in section 5 and are discussed in section 6.

2 Related Work

Though distributional similarity has widely been studied and has become an established method to find similar words, there is no consensus on the way the context of a word has to be defined and on the best way to compute the similarity between two contexts. In the most general definitions the context of a word consists of a number of words and their relation to the given word (Grefenstette, 1992; Curran and Moens, 2002). In the following we will only consider the simplest case in which there is only one relation: the relation of being in the same sentence. Each word can be represented by a so called *con-*

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 48–52, Atlanta, Georgia, June 14-15, 2013.

text vector in a high dimensional word space. Since these vectors will be sparse, often dimensionality reduction techniques are applied. In the present paper we use random indexing, introduced by Karlgren and Sahlgren (2001) and Sahlgren (2005) to reduce the size of the context vectors.

The way in which the context vectors are constructed also determines what similarity measures are suited. For random indexing Görnerup and Karlgren (2010) found that best results are obtained using L1-norm or Jensen-Shannon divergence (JSD). they also report that these measures highly correlate. We could confirm this in a preliminary experiment and therefore only use JSD in the following.

Recently, the question whether and how an appropriate context vector for a phrase can be derived from the context vectors of its components has become a central issue in distributional semantics (Clark and Pulman, 2007; Mitchell and Lapata, 2008; Widdows, 2008; Clarke et al., 2008). It is not yet clear which way of combining the vectors of the components is best suited for what goals. Giesbrecht (2010) and Mitchell and Lapata (2008) e.g. find that for noun-noun compounds the product of context vectors (corresponding to the intersection of contexts) and more complex tensor products give best results, while Guevara (2011) obtains best results for adjective-noun phrases with addition of vectors (corresponding to union of contexts). Since we do not (yet) have a single best similarity measure to distinguish definitions from non-definitions, we use a combination of similarity measures to train a model as e.g. also was done by Bär et al. (2012).

3 Frequency Dependency Correction of Jensen-Shannon Divergence

Weeds et al. (2004) observed that in tasks in which related words have to be found, some measures prefer words with a frequency similar to that of the target word while others prefer high frequent words, regardless of the frequency of the target word. Since Görnerup and Karlgren (2010) found that L1-norm and JSD give best results for similarity of random index vectors, we are especially interested in JSD. The JSD of two distributions p and q is given by

$$\text{JSD}(p,q) = \tfrac{1}{2}D(p||\tfrac{1}{2}p+\tfrac{1}{2}q)+\tfrac{1}{2}D(q||\tfrac{1}{2}p+\tfrac{1}{2}q) \quad (1)$$

where $D(p||q) = \Sigma_i p(i)\frac{\log p(i)}{\log q(i)}$ is the Kullback-Leibler divergence. We will follow the usual terminology of context *vectors*. However, we will always normalize the vectors, such that they can be interpreted as probability mass distributions. According to Weeds et al. (2004) the JSD belongs to the category of distance measures that tends to give small distances for highly frequent words. In Wartena et al. (2010) we also made this observation and therefore we added an additional constraint on the selection of keywords that should avoid the selection of too general words. In the present paper we try to explicitly model the dependency between the JSD and the number of occurrences of the involved words. We then use the difference between the JSD of the co-occurrence vectors of two words and the JSD expected on the base of the frequency of these words as a similarity measure. In the following we will use the dependency between the JSD and the frequency of the words directly. In (Wartena, 2013) we model the JSD instead as a function of the number of non zero values in the context vectors. The latter dependency can be modeled by a simpler function, but did not work as well with the SemEval data set.

Given two words w_1 and w_2 the JSD of their context vectors can be modeled as a function of the minimum of the number of occurrences of w_1 and w_2. Figure 3 shows the JSD of the context vectors of the words of the training set and the context vector of the definition phrase. In this figure the JSD of the positive and the negative examples is marked with different marks. The lower bound of the negative examples is roughly marked by a (red) curve, that is defined for context vectors c_1 and c_2 for words w_1 and w_2, respectively, by

$$\text{JSD}^{\text{exp}}(c_1,c_2) = a + \frac{1}{\hat{n}^b + c} \quad (2)$$

where $\hat{n} = \min(n(w_1), n(w_2))$ with $n(w)$ the number of occurrences of w in the corpus and with a, b and c constants that are estimated for each set of word pairs. For the pairs from the English training and test set the values are: $a = 0.15$, $b = 0.3$ and $c = 0.5$. Experiments on the training data showed that the final results are not very dependent on the exact values of these constants.

Finally, our new measure is simply defined by

$$\text{JSD}^{\text{norm}}(p,q) = \text{JSD}(p,q) - \text{JSD}^{\text{exp}}(p,q). \quad (3)$$

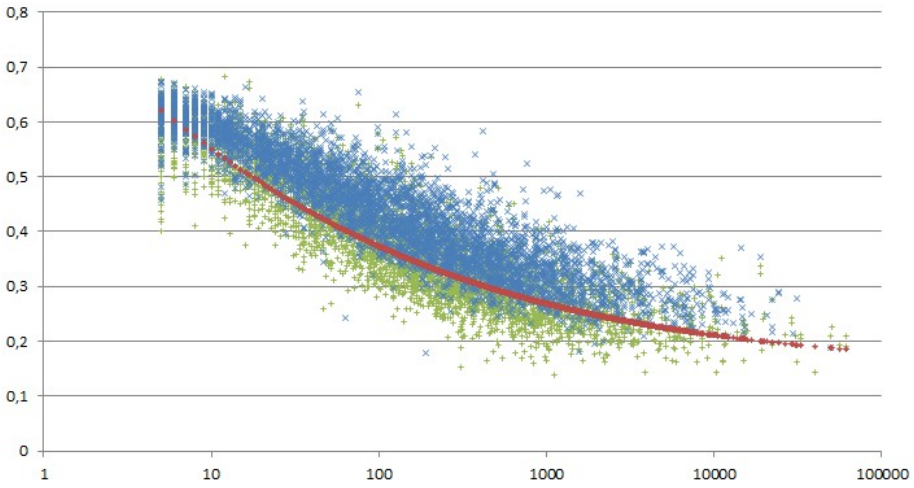

Figure 1: JSD (y-axis) of all pairs in the English training set versus the number of occurrences of the definition phrase (x-axis) in the UkWaC-Corpus. The positives examples are marked by a $+$, the negative examples by a $\times$. Most positive examples are hidden behind the negative ones. The solid (red) line gives the expected JSD.

4 System Description

The main assumption for our approach is, that a word and its definition are distributionally more similar than a word and an arbitrary definition. We use random indexing to capture distributional properties of words and phrases. Since similarity measures for random index vectors have biases for frequent or infrequent pairs, we use a combination of different measures. For the two-word definition phrases we can either estimate the context vector on the base of the two words that make up the phrase, or compute it directly from occurrences of the whole phrase in the corpus. The latter method has the advantage of being independent of assumptions about semantic composition, but might have the problem that it is based on a few examples only. Thus we use both distributions, and also include the similarities between the single word and each of the words of the definition.

4.1 Distributions

Consider a pair (w, d) with w a word and d a definition consisting of two words: $d = (d_1, d_2)$. Now for each of the words w, d_1, d_2 and the multiword d we compute context vectors using the random indexing technique. The context vectors are computed over the complete Wacky corpus. The context used for a word are all open-class words (i.e. Noun, Verb, Adjective, Adverb, etc. but not Auxiliary, Pronoun, etc.) in a sentence. Each word is represented by a random index vector of $10\,000$ dimensions in which 8 random positions have a non-zero value. The random vectors of all words in all contexts are summed up to construct context vectors (with length $10\,000$), denoted v_w, v_d, v_{d_1}, v_{d_2}. In many cases there are only very few occurrences of d, making the context vector v_d very unreliable. Thus we also compute the vectors $v_d^{\text{add}} = v_{d_1} + v_{d_2}$ and $v_d^{\text{mult}} = v_{d_1} \cdot v_{d_2}$. Finally, we also compute the general context vector (or background distribution) v_{gen} which is the context vector obtained by aggregating all used contexts.

4.2 Similarities

Table 1 gives an overview of the similarities computed for the context vector v_w. In addition we also compute $D(v_w||v_{\text{gen}})$, $D(v_d||v_{\text{gen}})$, $D(v_{d_1}||v_{\text{gen}})$, $D(v_{d_2}||v_{\text{gen}})$. The original intuition was that the definition of a word is usual given as a more general term or hypernym. It turned out that this is not the case. However, in combination with other features these divergences proved to be useful for the machine learning algorithm. Finally, we also use the direct (first-order) co-occurrence between w and d by computing the ratio between the probability with which we expect w and d to co-occur in one sentence if they would be independent, and the real probability of co-occurrence found in the corpus:

$$\text{co-occurrence-ratio}(w, d) = \frac{p(w, d)}{p(w) \cdot p(d)} \qquad (4)$$

Table 1: Similarity measures used to compute the similarity of a context vector of some word to various context vectors for a phrase $d = (d_1, d_2)$.

	v_d	v_{d_1}	v_{d_2}	v_d^{add}	v_d^{mult}
jsd	✓	✓	✓	✓	
jsd-norm	✓	✓	✓	✓	
cossim				✓	✓

Table 2: Results for English and German (no names dataset). Results on train sets are averaged results from 10-fold cross validation. Results on the test set are the official task results.

	AUC	Accuracy	F-Measure
Train English	0.88	0.80	0.79
Test English	-	0.80	0.79
Train German	0.90	0.83	0.82
Test German	-	0.83	0.81

where $p(w, d)$ is the probability that w and d are found in the same sentence, and $p(w)$, with w a word or phrase, the probability that a sentence contains w.

For the computation of $\text{JSD}^{\text{norm}}(v_w, v_d^{\text{add}})$ we need the number of occurrences on which v_d^{add} is based. As an estimate for this number we use $\max(n(d_1), n(d_2))$. The constants a, b and c in equation 2 are set to the following values: for all cases $a = 0.15$; for $\text{JSD}^{\text{norm}}(v_w, v_d)$ we let $b = 0.3$ and $c = 0.5$; for $\text{JSD}^{\text{norm}}(v_w, v_{d_1})$ and $\text{JSD}^{\text{norm}}(v_w, v_{d_2})$ we let $b = 0.35$ and $c = -0.1$; for $\text{JSD}^{\text{norm}}(v_w, v_d^{\text{add}})$ we let $b = 0.4$ and $c = -0.1$. For the German subtask $a = 0.28$ and slightly different values for b and c were used to account for slightly different frequency dependencies.

4.3 Combining Similarities

The 15 attributes for each pair obtained in this way are used to train a support vector machine (SVM) using LibSVM (Chang and Lin, 2011). Optimal parameters for the SVM were found by grid-search and 10-fold cross validation on the training data.

5 Results

In Table 2 the results are summarized. Since the task can also be seen as a ranking task, we include the Area Under the ROC-Curve (AUC) as a classical measure for ranking quality. We can observe that the results are highly stable between training set and test set and across languages. Table 3 gives the results that are obtained on the training set using one feature. We can observe that the normalized versions of the JSD always perform better than the JSD itself. Furthermore, we see that for the composed vectors the cosine performs better than the normalized JSD, while it performs worse than JSD for the other vectors (not displayed in the table). This eventually can be explained by the fact that we have to estimate the number of contexts for the calculation of jsd^{exp}.

Table 3: Results for English train set (average from 10-fold cross validation) using one feature

feature	Accuracy	AUC
$\text{jsd}(v_w, v_d)$	0.50	0.57
$\text{jsd}^{\text{norm}}(v_w, v_d)$	0.59	0.70
$\text{jsd}(v_w, v_{d_1})$	0.54	0.63
$\text{jsd}^{\text{norm}}(v_w, v_{d_1})$	0.61	0.69
$\text{jsd}(v_w, v_{d_2})$	0.57	0.65
$\text{jsd}^{\text{norm}}(v_w, v_{d_2})$	0.63	0.71
$\text{jsd}(v_w, v_d^{add})$	0.59	0.67
$\text{jsd}^{\text{norm}}(v_w, v_d^{add})$	0.66	0.74
$\text{cossim}(v_w, v_d^{add})$	0.69	0.76
$\text{cossim}(v_w, v_d^{mult})$	0.62	0.71
co-occ-ratio(w, d)	0.61	0.71

6 Conclusion

Though there are a number of ad-hoc decisions in the system the approach was very successful and performed best in the SemEval task on phrasal semantics. The main insight from the development of the system is, that there is not yet a single best similarity measure to compare random index vectors. The normalized JSD turns out to be a useful improvement of the JSD but is problematic for constructed context vectors, the formula in equation (2) is rather ad hoc and the constants are just rough estimates. The formulation in (Wartena, 2013) might be a step in the right direction, but also there we are still far away from a unbiased similarity measure with a well founded theoretical basis.

Finally, it is unclear, what is the best way to represent a phrase in distributional similarity. Here we use three different vectors in parallel. It would be more elegant if we had a way to merge context vectors based on direct observations of the phrase with a constructed context vector.

References

Daniel Bär, Chris Biemann, Iryna Gurevych, and Torsten Zesch. 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.

M. Baroni, S. Bernardini, A. Ferraresi, and E. Zanchetta. 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.

C.-C. Chang and C.-J. Lin. 2011. Libsvm : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27.

Stephen Clark and Stephen Pulman. 2007. Combining symbolic and distributional models of meaning. In *Proceedings of the AAAI Spring Symposium on Quantum Interaction*, pages 52–55.

Daoud Clarke, Rudi Lutz, and David Weir. 2008. Semantic composition with quotient algebras. In *Proceedings of the 9th International Conference on Computational Semantics (IWCS 2011)*.

James R. Curran and Marc Moens. 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. Association of Computational Linguistics.

Eugenie Giesbrecht. 2010. Towards a matrix-based distributional model of meaning. In *Proceedings of the NAACL HLT 2010 Student Research Workshop*, pages 23–28, Los Angeles, California. ACL.

Olaf Görnerup and Jussi Karlgren. 2010. Cross-lingual comparison between distributionally determined word similarity networks. In *Proceedings of the 2010 Workshop on Graph-based Methods for Natural Language Processing*, pages 48–54. ACL.

Gregory Grefenstette. 1992. Use of syntactic context to produce term association lists for text retrieval. In *SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 89–97. ACM.

Emiliano Guevara. 2011. Computing semantic compositionality in distributional semantics. In *Proceedings of the 9th International Conference on Computational Semantics (IWCS 2011)*, pages 135–144.

Jussi Karlgren and Magnus Sahlgren. 2001. From words to understanding. In *Foundations of Real-World Intelligence*, pages 294–308. CSLI Publications, Stanford, California.

Suresh Manandhar and Deniz Yuret, editors. 2013. *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantcis (*SEM 2013)*, Atlanta.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 236–244.

Magnus Sahlgren. 2005. An introduction to random indexing. In *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE*, volume 5.

Christian Wartena, Rogier Brussee, and Wouter Slakhorst. 2010. Keyword extraction using word co-occurrence. In *Database and Expert Systems Applications (DEXA), 2010 Workshop on*, pages 54–58. IEEE.

Christian Wartena. 2013. Distributional similarity of words with different frequencies. In *Proceedings of the Dutch-Belgian Information Retrieval Workshop*, Delft. To Appear.

Julie Weeds, David J. Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *COLING 2004, Proceedings of the 20th International Conference on Computational Linguistics*.

Dominic Widdows. 2008. Semantic vector products: Some initial investigations. In *Second Conference on Quantum Interaction, Oxford, 26th 28th March 2008*.