Automatic Annotation of Co-Occurrence Relations

Dirk Goldhahn[1,2], Uwe Quasthoff[1]

[1]University of Leipzig
Department of Computer Science, NLP Group
Johannisgasse 26, 04103 Leipzig, Germany

[2]Max Planck Institute for Human Cognitive and Brain Sciences
Department of Neurophysics
Stephanstraße 1A, 04103 Leipzig, Germany

E-mail: goldhahn@cbs.mpg.de, quasthoff@informatik.uni-leipzig.de

Abstract

We introduce a method for automatically labelling edges of word co-occurrence graphs with semantic relations. Therefore we only make use of training data already contained within the graph. Starting point of this work is a graph based on word co-occurrence of the German language, which is created by applying iterated co-occurrence analysis. The edges of the graph have been partially annotated by hand with semantic relationships. In our approach we make use of the commonly appearing network motif of three words forming a triangular pattern. We assume that the fully annotated occurrences of these structures contain information useful for our purpose. Based on these patterns rules for reasoning are learned. The obtained rules are then combined using Dempster-Shafer theory to infer new semantic relations between words. Iteration of the annotation process is possible to increase the number of obtained relations. By applying the described process the graph can be enriched with semantic information at a high precision.

1. Introduction

In this paper we introduce ways to automatically label edges of word co-occurrence graphs with semantic relations. To achieve this a training set of semantically annotated relations is used. Based on the patterns of these relationships in the training set, rules for reasoning are learned. The rules are then used to annotate new edges.

Basis of this work is a co-occurrence graph of the German language consisting of more than 9 million words. It was created in the project Deutscher Wortschatz (German vocabulary), www.wortschatz.uni-leipzig.de. The nodes of this graph are words which are connected if they occur significantly often together within sentences. The graph used here is created by applying the co-occurrence analysis again, this time to the above co-occurrence graph. In this iterated co-occurrence graph two words are connected if they have a lot of common neighbors, i.e. if there are a lot of words having both words as ordinary co-occurrences (Heyer, 2006). The iteration increases the portion of paradigmatic relations. This way more pairs with similar contexts can be found and this gives rise to a higher percentage of pairs of words with a classical semantic relationship (Biemann, 2004).

On this basis a process of computer aided manual annotation was executed (Biemann, 2005). Some edges of the graph were labeled with semantic relations (like: synonym, cohyponym, hypernym, typical-feature as a relation between adjectives and nouns, typical-object-of as relation between nouns and verbs, part-of as relation between nouns). The relations used were inspired by classical semantics and the data in the graph.

All in all about 400.000 edges have been annotated. The most prominent relations are shown in table 1.

Relation	POS word 1	POS word 2	Quantity
cohyponymy	noun	noun	98953
hypernymy	noun	noun	35902
hyponymy	noun	noun	35890
synonymy	noun	noun	31130
typical feature of	adjective	noun	17459
has typical feature	noun	adjective	17458
has typical object	verb	noun	15199
typical object of	noun	verb	15180
cohyponymy	adjective	adjective	12808
synonymy	verb	verb	9334
part/material of	noun	noun	8260
has part/material	noun	noun	8240
hypernymy	verb	verb	8226
hyponymy	verb	verb	8224
synonymy	adjective	adjective	7652
has typical location	noun	noun	7028
typical location for	noun	noun	7004
cohyponymy	verb	verb	5889
typical activity of	verb	noun	5726
typical subject of	noun	verb	5723
proper name of	noun	noun	3659
antonymy	adjective	adjective	3138

Table 1: The most prominent annotated relations and the number of occurrences in the graph. Column two and three represent the part of speech of the involved words.

2. Automatic Annotation

In this paper we introduce The construction of this graph by hand was very costly and time consuming. But still it suffers from sparse coverage. Making it denser this way would only be possible with great effort. Another option to obtain more semantic information is to automate the annotation process. In the following we will present a feasible approach to extend the graph automatically.

Because of the high costs when doing it manually, there have already been several projects which have dealt with the automatic creation or extension of semantic networks. Most of these approaches use lexico-syntactic patterns extracted from large corpora to infer relations between words. The patterns are either created by hand or inferred automatically. Work in this area includes the inference of hyponymy and other relations (Hearst, 1992), building a noun hierarchy from text (Caraballo, 1999) and reasoning meronymy (Girju, 2003), synonymy (Lin, 2003) and verb relations (Chklovski, 2004). Others infer relations like hyponymy while incorporating sense disambiguation and globally optimizing the entire structure of the taxonomy (Snow, 2005, 2006).

In this approach we utilize only information contained within the graph when reasoning new relations. The idea is to use a training set of annotated edges to create rules which can be applied later to infer further edges of the graph. The simplest rule can be found as follows: Assume we have three words A, B and C in the training set which form a triangle in the graph and all edges of the triangle are labeled with synonymy. If only two of the edges were marked with synonymy and no mark at the third edge, we could infer synonymy for the third edge by transitivity. This rule generation can be generalized as follows: Having a triangle of three words A, B and C with edges annotated with relations R1, R2 and R3, we produce a rule which implies relation R3 for a non-annotated edge if the other edges are labeled with R1 and R2. Such a rule can be sound and effective as the synonymy rule above or very poor. But the correctness of such a rule can be estimated by testing it on the fully annotated training set. Only rules with a correctness above some threshold will be used later.

3. Properties of the Graph

The triangular structures used for the creation of rules to infer new edges were chosen depending on the structure of the graph.

Semantic networks like the underlying co-occurrence graph commonly have scale-free and small world properties (Newman, 2003; Steyvers, 2005). In the following we will show that the selected method for manual annotation leads to an annotated graph whose analysis also reveals the necessary properties (Clauset, 2009) of small worlds and scale-freeness.

Scale-freeness can easily be shown by looking at the distribution of the node degrees of the graph. In scale-free networks this distribution follows roughly a power-law. In figure 1 the node degrees of the annotated graph are plotted with logarithmically scaled axes. As expected the data points are nearly forming a straight line in this scaling, indicating the power law.

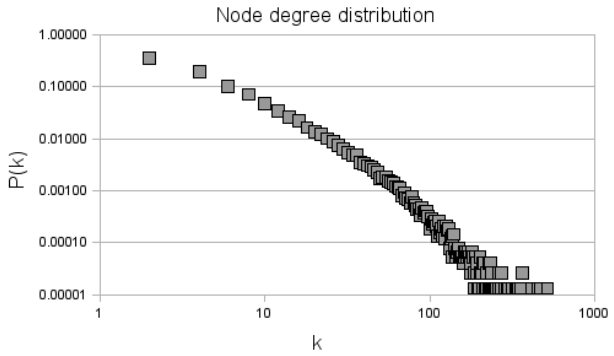

Figure 1: Node degree distribution of the annotated co-occurrence graph. The axes are logarithmically scaled and labeled with node degree k and probability P(k).

Important features of the small world phenomenon are a high local clustering coefficient and short average path length even in sparse graphs. The annotated co-occurrence graph only has an average node degree of 9.72. But still 75% of all nodes can be reached within 5 steps from a central node of the network, which indicates a short average path length. A high local clustering coefficient can also be shown for this graph. This property means that two neighbors of a node are often connected among themselves. Although the annotated co-occurrence graph is sparse, it has a high local clustering coefficient of about 0.21. So the graph forms many triangular structures of three nodes (for an example see figure 2).

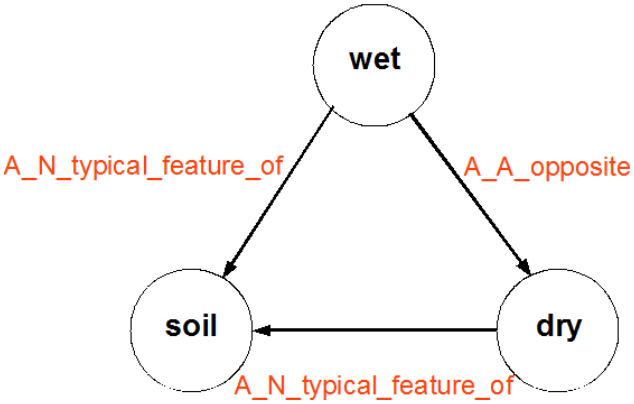

Figure 2: Example of a subgraph of the annotated co-occurrence graph consisting of three nodes with fully annotated edges.

Because of the high number of these small informative motifs they are selected for the creation of rules to infer new edges. So the local semantic information of the edges of all triangular network motifs with three annotated edges is used to create rules for the reasoning of relations. Figure 3 shows some examples of these rules.

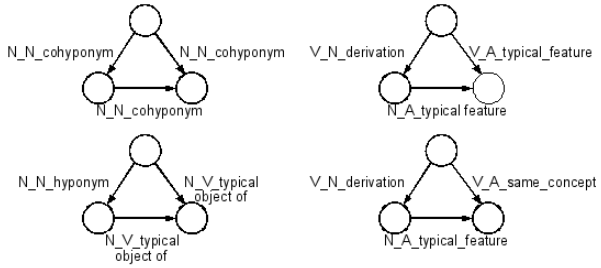

Figure 3: Examples of rules. The edges are labeled with the required part of speech of the nodes and with a relation. The two upper edges are known and the third edge is annotated with its most likely relation.

4. Combination of Rules – Dempster-Shafer Theory

Because of the sparse annotation of the graph most rules are very "weak". This means that there is a high amount of uncertainty in them because of the high number of edges without annotation. To raise the plausibility of inferred relations a method for the combination of rules is needed that also takes into account the mentioned uncertainty. This way supporting rules as in table 2 can boost the annotation probability of edges that are part of several triangular structures as shown in figure 4.

First Relation	Second Relation	Likeliest Third Relation
typical feature of	opposite	typical feature of
typical feature of	cohyponym	typical feature of

Table 2: Two rules fitting the triangles of figure 4.

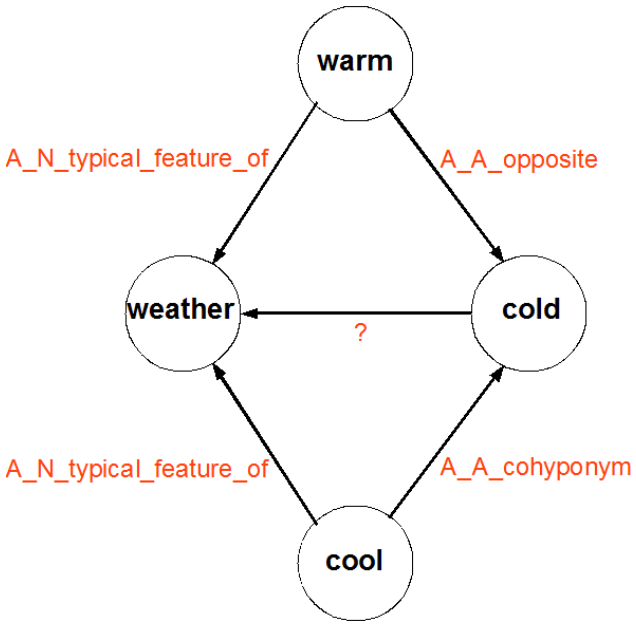

Figure 4: Example of a not yet labeled edge which is part of two otherwise fully annotated triangles.

In this work Dempster-Shafer theory, also known as evidence theory, is used to achieve this (Shafer, 1976; Barnett, 1981).

This theory is a generalization of the Bayesian probability theory and is used in fields like pattern recognition to combine propositions from different sources. In this framework the classical probability axioms do not hold, especially:

$p(A)=1-p(\overline{A})$

is not always fulfilled.

Instead of a normal probability a two-dimensional measure is used. It consists of degree of belief and plausibility, which form a probability range. Degree of belief *Bel(A)* is a measure expressing the certainty that a proposition *A* will happen. Plausibility *Pl(A)* of a proposition *A* on the other hand is the certainty with which one cannot rule out the possibility that *A* will happen. This way uncertainty of statements or rules is expressible. Figure 5 illustrates the basic assumptions of Dempster-Shafer theory compared to classical theory of probability.

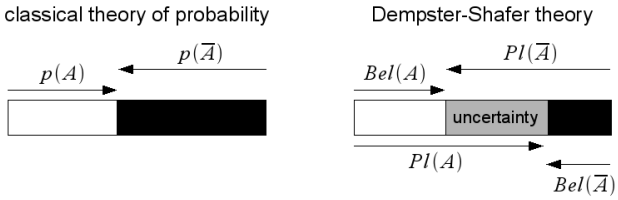

Figure 5: Comparison of key features of classical probability theory and Dempster-Shafer theory.

The basis for the computation of the mentioned measures is a basic probability number m also referred to as believe mass. The theory of evidence assigns a belief mass to each element of the power set of the propositions. The basic probability number *m(A)* represents the exact belief in proposition *A*. All belief masses of a set of mass assignments add up to a total of 1.

The degree of belief *Bel(A)* of a proposition A is then calculated from these basic probability numbers. *Bel(A)* equals the sum of the basic probability numbers of all propositions *B* that are a subset of *A*,

$$Bel(A)=\sum_{B\subseteq A} m(B).$$

The plausibility of a proposition A is the sum of the basic probability numbers of all propositions *B* whose intersection with *A* is not empty,

$$Pl(A)=\sum_{A\cap B\neq\emptyset} m(B).$$

Degree of belief and plausibility of a proposition *A* and its complementary proposition are connected as follows:

$Pl(A)=1-Bel(\overline{A})$.

Now all that is needed to apply this theory is a means for combining independent sets of mass assignments m_1 and m_2. This would allow us to combine evidence from difference sources. In evidence theory Dempster's rule is used to achieve this. This rule combines the belief masses in the following way:

$$m_1 \oplus m_2(A)=\frac{\sum_{B_1\cap B_2=A} m_1(B_1)\cdot m_2(B_2)}{1-\sum_{B_1\cap B_2=\emptyset} m_1(B_1)\cdot m_2(B_2)}.$$

This process can be repeated several times if more than two sources have to be integrated. The combined belief masses can then be used to calculate the degree of belief and plausibility of certain propositions.

5. The process of annotation

The mechanisms that have been introduced in the previous sections will now be used on the annotated co-occurrence graph to infer semantic relations.

In a first step the rules, as described in section 2 and 3, are created. All fully annotated triangles are used for their generation. For every possible combination of two relations A and B that also appears in the graph the occurrences of different relations C on the third edge of the triangle are counted. Third edges that have not yet been annotated are also taken into account to have a measure of uncertainty for each rule. Having counted all these numbers it is possible to assign belief masses to all potential relations of the third edge. This way 814 rules are created.

Next edges without a relation are labeled. This can only be done if the edge is part of a triangle with two annotated relations. In this step we only investigate edges that are part of at least two of such structures since single rules normally do not have enough validity. Dempster-Shafer theory is then utilized to combine the respective rules and to calculate degree of belief and plausibility of the likeliest relation.

This way nearly 300.000 edges can be annotated with new relations. By introducing a threshold for annotation based on degree of belief and plausibility it is possible to accept only those semantic relations that are very likely to be annotated correctly. By iterating the process of annotation more edges can be generated and the graph can be densified even more. Figure 6 illustrates the numbers of new relations that can be achieved. In this case only the relations above a threshold of about 0.5 were accepted after each step. It is salient that the number of edges that can be annotated increases with each iteration.

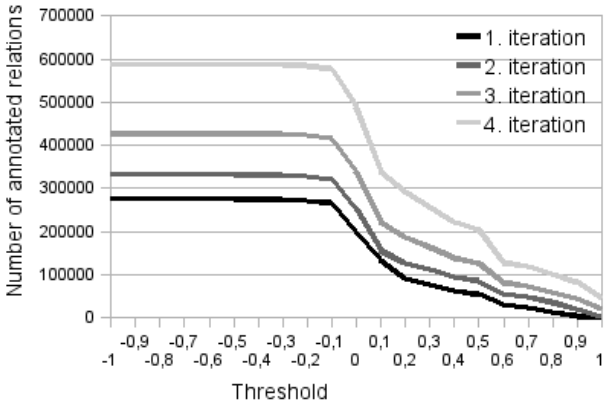

Figure 6: Iteration of the annotation. The chart shows the number of relations the algorithm is able to annotate in each iteration. Only new relations above a certain threshold (~0.5 in this example) are accepted and are involved in the reasoning of new relations in the next steps of the iteration. The threshold is based on degree of belief and plausibility of the annotated relation.

6. Evaluation of the algorithm

To validate the results of this approach about 10% of the relations are removed from the graph as a testing set before rule creation and annotation are started. Again only edges that are part of at least two partially labeled triangles are annotated with semantic relations. The results are then validated against the relations the respective edges were manually annotated with before the testing set was removed from the graph. Precision and recall are measured dependent on a threshold composed of degree of belief and plausibility of the annotation of each edge. The results are visualized in figure 7. Obviously a high precision can be reached but there is a tradeoff for recall. When the process is iterated the precision at a fixed recall decreases with each step of the iteration.

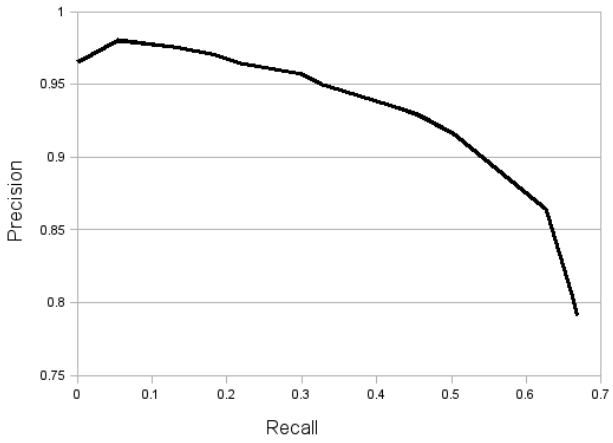

Figure 7: Precision and recall of the re-annotation of edges (subject to a threshold for annotation) after removing about 10% of the original graph as a testing set.

In addition to the 370.000 edges that have been annotated manually with relations, about 56.000 nodes of the graph have been labeled with semantic primitives, such as natural, artificial, alive, proper name, place or condition. An example can be seen in figure 8.

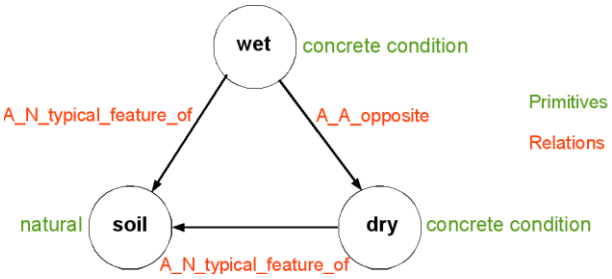

Figure 8: The subgraph already depicted in figure 2. Semantic primitives are added to the nodes.

These primitives can be incorporated into the process of automatic annotation. Therefore the process of rule creation is modified to include the information present in the semantic primitives of many nodes of the graph. The results of this approach are shown in figure 9. It is clearly visible that the incorporation of the semantic properties of the words improves the correctness of the annotation.

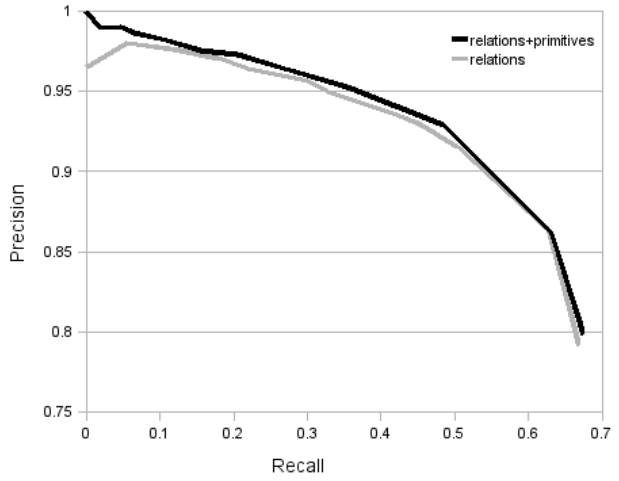

Figure 9: Changes in precision and recall when incorporating semantic primitives into the algorithm.

It is important to mention that the algorithm presented here is in one way limited in its capabilities. The method is able to label new edges in areas of the graph where many annotated edges already exist. So especially the density of annotations in clusters of the network will be increased by our method. But a node that is not connected to other nodes by semantic relations will never receive an annotation for any of its edges.

7. Conclusion

We have presented an algorithm for automatically annotating a graph with sparse semantic information with further semantic relations. By learning rules that utilize the distribution of patterns of semantic relations already present in the graph this can be achieved at a high precision.

In the future this approach could be enhanced by taking into account other features [see e.g. Biemann 2005] than the local semantic motifs for rule learning and reasoning. Since the mechanism is of statistical nature, it could be applied to other semantic networks and other languages without problems. The algorithm also needs to be more thoroughly evaluated, especially a comparison to other methods for the combination of probability distributions (Turney, 2003) would be useful. Last but not least an application to semantic networks with disambiguated senses would be possible since it does not matter if the nodes of the graph are words or senses.

8. References

Barnett, J.A. (1981). Computational Methods for A Mathematical Theory of Evidence. In Proceedings of the Seventh International Joint Conference on Artificial Intelligence. pp. 868-875.

Biemann, C.; Bordag, S.; Quasthoff, U. (2004). Automatic Acquisition of Paradigmatic Relations using Iterated Co-occurrences. In Proc. LREC2004. Lissabon, Portugal.

Biemann, C. (2005). Semantic Indexing with Typed Terms Using Rapid Annotation. In Proc. TKE-05-Workshop on Methods and Applications of Semantic Indexing. Copenhagen, Denmark.

Caraballo, S.A. (1999). Automatic construction of a hypernym-labeled noun hierarchy from text. In Proc. 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics. pp. 120-126.

Chklovski, T.; Pantel, P. (2004). VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In Proc. EMNLP-2004.

Clauset, A.; Shalizi, C.R.; Newman, M.E.J. (2009). Power-Law Distributions in Empirical Data. SIAM Review 51. pp. 661-703.

Girju, R.; Badulescu, A.; Moldovan, D. (2003). Learning Semantic Constraints for the Automatic Discovery of Part-Whole Relations. In Proc. HLT-03.

Hearst, M. (1992). Automatic Acquisition of Hyponyms from Large Text Corpora. In Proc. COLING-92.

Heyer, G.; Quasthoff, U.; Wittig, T. (2006). Text Mining: Wissensrohstoff Text, Konzepte, Algorithmen, Ergebnisse. Witten, Germany.

Lin, D.; Zhao, S.; Qin, L.; Zhou, M. (2003). Identifying Synonyms among Distributionally Similar Words. In Proc. IJCAI-03.

Newman, M.E.J.(2003). The structure and function of complex networks. SIAM Review 45. pp. 167-256.

Shafer, G. (1976). A Mathematical Theory of Evidence. Princeton, USA.

Snow, R.; Jurafsky, D.; Ng, A.Y. (2005). Learning syntactic patterns for automatic hypernym discovery. In NIPS 17.

Snow, R.; Jurafsky, D.; Ng, A.Y. (2006). Semantic taxonomy induction from heterogenous evidence. In Proc. COLING/ACL2006.

Steyvers, M.; Tenenbaum, J. (2005). The Large-Scale Structure of Semantic Networks. Statistical Analyses and a Model of Semantic Growth. In Cognitive Science, Volume 29, Number 1.

Turney, P.;Littman, M.; Bigham, J.; Shnayder, V. (2003). Combining independent modules to solve multiple-choice synonym and analogy problems. In Proc. RANLP-2003. pp. 482–489.