

# Recognition of synonyms by a lexical graph

Peter Siniakov

siniakov@inf.fu-berlin.de

Database and Information Systems Group, Freie Universität Berlin  
Takustr. 9, 14195 Berlin, Germany

## Abstract

Semantic relationships between words comprised by thesauri are essential features for IR, text mining and information extraction systems. This paper introduces a new approach to identification of semantic relations such as synonymy by a lexical graph. The graph is generated from a text corpus by embedding syntactically parsed sentences in the graph structure. The vertices of the graph are lexical items (words), their connection follows the syntactic structure of a sentence. The structure of the graph and distances between vertices can be utilized to define metrics for identification of semantic relations. The approach has been evaluated on a test set of 200 German synonym sets. Influence of size of the text corpus, word generality and frequency has been investigated. Conducted experiments for synonyms demonstrate that the presented methods can be extended to other semantic relations.

## 1 Introduction

Once predominantly used by human authors to improve their style avoiding repetitions of words or phrases, thesauri now serve as an important source of semantic and lexical information for automatic text processing. The electronic online thesauri such as WordNet (2005) and OpenThesaurus (2005) have been increasingly employed for many IR and NLP problems. However, considerable human effort is required to keep up with the evolving language and many subdomains are not sufficiently covered (Turney, 2001). Many domain-specific words or word senses are not included;

inconsistency and bias are often cited as further major deficiencies of hand-made thesauri (Curran and Moens, 2002), (Senellart and Blondel, 2003). There is a continuous demand for automatic identification of semantic relations and thesaurus generation. Such tools do not only produce thesauri that are more adapted to a particular application in a certain domain, but provide also assistance for lexicographers in manual creation and keeping the hand-written thesauri up to date. Numerous applications in IR (e.g. query expansion) and text mining (identification of relevant content by patterns) underline their usefulness.

## 2 Related work

Identification of semantic relations has been approached by different communities as a component of a knowledge management system or application of a developed NLP framework. Many approaches are guided by the assumption that similar terms occur in similar context and obtain a context representation of terms as attribute vectors or relation tuples (Curran and Moens, 2002), (Ruge, 1997), (Lin, 1998). A similarity metric defined on the context representations is used to cluster similar terms (e.g. by the nearest neighbor method). The actual definitions of context (whole document (Chen and Lynch, 1992), textual window, some customized syntactic contexts, cf. (Senellart and Blondel, 2003)) and similarity metric (cf. (Manning and Schütze, 1999), (Curran and Moens, 2002)) are the essential distinguishing features of the approaches.

A pattern-based method is proposed by Hearst (Hearst, 1998). Existing relations in the WordNet database are used to discover regular linguistic patterns that are characteristic for these relations. The patterns contain lexical and syntactic elements and are acquired

from a text corpus by identifying common context of word pairs for which a semantic relation holds. Identified patterns are applied to a large text corpus to detect new relations. The method can be enhanced by applying filtering steps and iterating over new found instances (Phillips and Riloff, 2002).

Lafourcade and Prince base their approach on reduction of word semantics to conceptual vectors (vector space is spanned by a hierarchy of concepts provided by a thesaurus, (Lafourcade, 2001)). Every term is projected in the vector space and can be expressed by the linear combination of conceptual vectors. The angle between the vectorial representations of two terms is used in calculation of thematic closeness (Lafourcade and Prince, 2001). The approach is more closely related to our approach since it offers a quantitative metric to measure the degree of synonymy between two lexical items.

In contrast, Turney (Turney, 2001) tries to solve a quite simpler “TOEFL-like” task of selecting a synonym to a given word from a set of words. Mutual information related to the co-occurrence of two words combined with information retrieval is used to assess the degree of their statistical independency. The least independent word is regarded synonymous.

Blondell et al. (Blondel et al., 2004) encode a monolingual dictionary as a graph and identify synonyms by finding subgraphs that are similar to the subgraph corresponding to the queried term.

The common evaluation method for similarity metrics is comparing their performance on the same test set with the same context representations with some manually created semantic source as the gold standard (Curran and Moens, 2002). Abstracting from results for concrete test sets, Weeds et al. (2004) try to identify statistical and linguistic properties on that the performance of similarity metrics generally depends. Different bias towards words with high or low frequency is recognized as one reason for the significant variance of k-nearest neighbors sets of different similarity metrics.

### 3 Construction of the lexical graph

The assumption that similar terms occur in similar context leads to the establishing of ex-

PLICIT context models (e.g. in form of vectors or relation tuples) by most researchers. We build an implicit context representation connecting lexical items in a way corresponding to the sentence structure (as opposed to (Blondel et al., 2004)), where a term is linked to every word in its definition). The advantage of the graph model is its transitivity: not only terms in the immediate context but also semantically related terms that have a short path to the examined term (but perhaps have never occurred in its immediate context) can contribute to identification of related terms. The similarity metric can be intuitively derived from the distance between the lexical vertices in the graph.

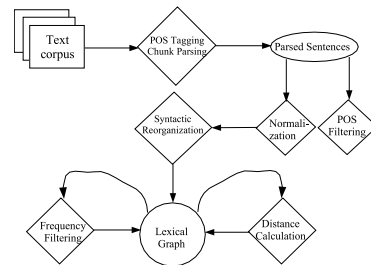


Figure 1: Main steps during graph construction

To construct the lexical graph articles from five volumes of two German computer journals have been chunk-parsed and POS tagged using *TreeTagger* (2004). To preserve the semantic structure of the sentences during the graph construction, i.e. to connect words that build the actual statement of the sentence, parsed sentences are preprocessed before being inserted in the graph (fig. 1). The punctuation signs and parts of speech that do not carry a self-contained semantics (such as conjunctions, pronouns, articles) are removed in a POS filtering step. Tokenization errors are heuristically removed and the words are replaced by their normal forms (e.g. infinitive form for verbs, nominative singular for nouns). German grammar is characterized by a very frequent use of auxiliary and modal verbs that in most cases immediately precede or follow the semantically related sentence parts such as direct object or prepositional phrase while the main verb is often not adjacent to the related parts in a sentence. Since the direct edge between the main verb and non-adjacent related sentence parts cannot be drawn, the

sentence is syntactically reorganized by replacing the modal or auxiliary verbs by the corresponding main verb. Another syntactic rearrangement takes place when detachable prefixes are attached to the corresponding main verb. In German some prefixes of verbs are detached and located at the end of the main clause. Since verbs without a prefix have a different meaning prefixes have to be attached to the verb stem. The reorganized sentence

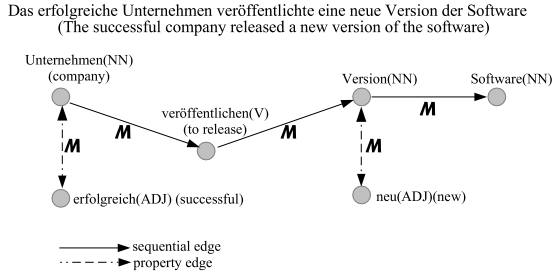


Figure 2: An example of a sentence transformed in a lexical graph

can be added to the graph inserting the normalized words in a sentence as vertices and connecting the adjacent words by a directed edge. However, some adjacent words are not semantically related to each other, therefore the lexical graph features two types of edges (see an example in fig. 2). A *property edge* links the head word of a syntactic chunk (verb or noun phrase) with its modifiers (adverbs or adjectives respectively) that characterize the head word and is bidirectional. A *sequential edge* connects the head words (e.g. main verbs, head nouns) of syntactic chunks reflecting the “semantic backbone” of the sentence.

The length of an edge represents how strong two lexical items are related to each other and depends therefore on the frequency of their co-occurrence. It is initialized with a maximum length  $M$ . Every time an existing edge is found in the currently processed sentence, its current length  $CurLen$  is modified according to  $CurLen = \frac{M}{\frac{M}{CurLen} + 1}$ ; hence the length of an edge is inversely proportional to the frequency of co-occurrence of its endpoints.

After all sentences from the text corpus have been added to the lexical graph, vertices (words) with a low frequency ( $\leq \theta$ ) are removed from the graph to primarily accelerate the distance calculation. Such rarely oc-

curing words are usually proper nouns, abbreviations, typos etc. Because of the low frequency semantic relations for these words cannot be confidently identified. Therefore removing such vertices reduces the size of the graph significantly without performance penalty (the graph generated from 5 journal volumes contained ca. 300000 vertices and 52191 after frequency filtering with  $\theta = 8$ ). Experimental results feature even a slightly better performance on filtered graphs. To preserve semantic consistency of the graph and compensate removal of existing paths the connections between the predecessors and successors of removed vertices have to be taken into account: the edge length  $e(p, s)$  between the predecessor  $p$  to the successor  $s$  of the removed vertex  $r$  can incorporate the length of the path  $length(p, r, s)$  from  $p$  to  $s$  through  $r$  by calculating the halved harmonic mean:  $e(p, s) = \frac{e(p, r) * l_{prs}}{e(p, r) + l_{prs}}$ .  $e(p, s)$  is the more reduced the smaller  $length(p, r, s)$  is and if they are equal,  $e(p, s)$  is half as long after merging.

Beside direct edges an important indication of semantic closeness is the distance, i.e. the length of the shortest path between two vertices. Distances are calculated by the Dijkstra algorithm with an upper threshold  $\Theta$ . Once the distances from a certain vertex reach the threshold, the calculation for this vertex is aborted and the not calculated distances are considered infinite. Using the threshold reduces the runtime and space considerably while the semantic relation between the vertices with distances  $> \Theta$  is negligible.

The values of  $M$ ,  $\theta$  and  $\Theta$  depend on the particular text corpus and are chosen to keep the size of the graph feasible.  $\theta$  can be determined experimentally incrementing it as long as the results on the test set are improving. The resulting graph generated from five computer journals volumes with  $M = 2^{20}$ ,  $\theta = 8$ ,  $\Theta = 60000$  contained 52191 vertices, 4,927,365 edges and 376,000,000 distances.

#### 4 Identification of synonyms

The lexical graph is conceived as an instrument to identify semantic relations such as synonymy and hypernymy between lexical items represented by its vertices. The main

focus of our research was finding synonyms albeit some results can be immediately transferred for identification of hyponyms. To provide a quantitative measure of synonymy different similarity metrics were defined on the lexical graph. Given a word, the system uses the metric to calculate the closest vertices to the vertex that represents this word. The result is a ranked list of words sorted by the degree of synonymy in descending order. Every metric  $sim$  is normalized to be a probability measure so that given a vertex  $v_i$  the value  $sim(v_i, v_j)$  can be interpreted as the probability of  $v_j$  being synonym to  $v_i$ . The normalization is performed for each metric  $sim$  by the following functions:

$$n_{min}(sim(v_i, v_j)) = \frac{\min(sim(v_i, v_1), \dots, sim(v_i, v_n))}{sim(v_i, v_j)}$$

for metrics that indicate maximum similarity to a vertex  $v_i$  by a minimum value and

$$n_{max}(sim(v_i, v_j)) = \frac{sim(v_i, v_j)}{\max(sim(v_i, v_1), \dots, sim(v_i, v_n))}$$

for metrics that indicate maximum similarity to a vertex  $v_i$  by a maximum value, where  $v_1 \dots v_n$  are the set of graph vertices. In both cases the top-ranked word has the maximum likelihood of 1 to be a synonym of  $v_i$ . The normalized ranked lists are used for the comparison of different metrics and the evaluation of the approach (see sec. 5).

A similarity metric is supposed to assess the semantic similarity between two vertices of the lexical graph. Since the distance metric *DistanceM* used for calculation of distances between the vertices in the graph indicates how semantically related two vertices are, it can be used as a similarity metric. As the graph is directed, the distance metric is asymmetric, i.e. the distance from  $v_j$  to  $v_i$  does not have to be equal to the distance from  $v_i$  to  $v_j$ . The major drawback of the *DistanceM* is that it takes into account only one path between the examined vertices. Even though the shortest path indicates a strong semantic relation between the vertices, it is not sufficient to conclude synonymy that presupposes similar word senses.

Therefore more evidence for strong semantic relation with the particular aspect of similar word senses should be incorporated in

the similarity metric. The *property neighbors* of a vertex  $v_i$  (adjacent vertices connected with  $v_i$  by the property edge) play significant role in characterizing similar senses. If two terms share many characteristic properties, there is a strong evidence of their synonymy. A shared property can be regarded as a *witness* of the similarity of two word senses. There are other potential witnesses, e.g. transitive verbs shared by their direct objects; however, we restricted this investigation to the property neighbors as the most reliable witnesses.

The simple method to incorporate the concept of the witnesses into the metric is to determine the number of common property neighbors:

$$NaivePropM(v_i, v_j) = |prop(v_i) \cap prop(v_j)|$$

where  $prop(v_i) = \{v_k | e(i, k) \text{ is a property edge}\}$ . This method disregards, however, the different degree of correlation between the vertices and their property neighbors that is reflected by the length of property edges. A property is the more significant, the stronger the correlation between the property and the vertex is, that is the shorter the property edge is. The degree of synonymy of two terms depends therefore on the number of common properties and the lengths of paths between these terms leading through the properties. Analogously to the electric circuit one can see the single paths through different shared properties as channels in a parallel connection and path lengths as "synonymy resistances". Since a bigger number of channels and smaller single resistances contribute to the decreasing of the total resistance (i.e. the evidence of synonymy increases), the idea of *WeiPropM* metric is to determine the similarity value analogously to the total resistance in a parallel connection:

$$WeiPropM'(v_i, v_j) = \left( \sum_{k=1}^n \frac{1}{length(v_i, p_k, v_j)} \right)^{-1}$$

where  $length(v_i, p_k, v_j) = e(v_i, p_k) + e(p_k, v_j)$  is the length of the path from  $v_i$  to  $v_j$  through  $p_k$  and  $p_k \in prop(v_i) \cap prop(v_j)$ .

Another useful observation is that some properties are more valuable witnesses than the others. There are very general properties that are shared by many different terms and

some properties that are characteristic only for certain word senses. Thus the number of property neighbors of a property can be regarded as a measure of its quality (in the sense of characterizing the specific word meaning). *WeiPropM* integrates the quality of a property by weighting the paths leading through it by the number of its property neighbors:

$$WeiPropM(v_i, v_j) = \left( \sum_{k=1}^n \frac{1}{(e(v_i, p_k) + e(p_k, v_j)) * |prop(p_k)|} \right)^{-1}$$

where  $p_k \in prop(v_i) \cap prop(v_j)$ .

*WeiPropM* measures the correlation between two terms based on the path lengths. Frequently occurring words tend to be ranked higher because the property edge lengths indirectly depend on the absolute word frequency. Because of high absolute frequency of words the frequency of their co-occurrence with different properties is generally also higher and the property edges are shorter. Therefore to compensate this deficiency (i.e. to eliminate the bias discussed in (Weeds et al., 2004)) an edge length from a property to a ranked term  $e(p_k, v_j)$  is weighted by the square root of its absolute frequency  $\sqrt{freq(v_j)}$ . Using the weighted edge length between the property and the ranked term we cannot any longer calculate the path length between  $v_i$  and  $v_j$  as the sum  $length(v_i, p_k, v_j) = e(v_i, p_k) + e(p_k, v_j) * \sqrt{freq(v_j)}$  because the multiplied second component significantly outweighs the first summand. Relative path length can be used instead where both components are adequately taken into account and added relatively to the minimum of the respective component: let  $min1$  be  $min(e(v_i, p_a), \dots, e(v_i, p_n))$  where  $p_k \in prop(v_i)$  and  $min2 = min(\dots, e(p_k, v_j) * \sqrt{freq(v_j)}, \dots)$  where  $p_k \in prop(v_i) \cap prop(v_j)$ . Relative path length would be  $\frac{e(v_i, p_k)}{min1} + \frac{e(p_k, v_j) * \sqrt{freq(v_j)}}{min2}$ . Further experimental observation suggests that when searching for synonyms of  $v_i$  the connection between  $v_i$  and the property is more significant than the second component of the path – the connection between the property and the ranked term  $v_j$ . Therefore when calculating the relative path length the first component

has to be weighted stronger (the examined ratio was 2:1). The corresponding metric can be defined as follows:

$$FirstCompM(v_i, v_j) = \left( \sum_{k=1}^n \frac{1}{RelPathLength(k) * \sqrt{|prop(p_k)|}} \right)^{-1}$$

where  $RelPathLength(x) =$

$$\frac{2}{3} * \frac{e(v_i, p_x)}{min1} + \frac{1}{3} * \frac{e(p_x, v_j) * \sqrt{freq(v_j)}}{min2}$$

As opposed to *NaivePropM* and *WeiPropM* *FirstCompM* is not symmetric because of the emphasis on the first component.

## 5 Experiments

For evaluation purposes a test corpus of 200 synonym sets was prepared consulting (OpenThesaurus, 2005). The corpus consists of 75 everyday words (e.g. “Präsident” (president), “Eingang” (entrance) “Gruppe” (group)), 60 abstract terms (e.g. “Ursache” (reason), “Element”, “Merkmal” (feature)) and 65 domain-specific words (e.g. “Software”, “Prozessor” (CPU)). The evaluation strategy is similar to that pursued in (Curran and Moens, 2002). The similarity metrics do not distinguish between different word senses returning synonyms of all senses of the polysemous words in a single ranked list. Therefore the synonym set of a word in the test corpus is the union of synonym sets of its senses. To provide a measure for overall performance and to compare the different metrics a function measuring the *similarity score* (*SimS*) was defined that assigns a score to a metric for correctly found synonyms among the 25 top-ranked. The function assigns 25 points to the correctly found top-ranked synonym of  $v_i$  ( $SimS(0, v_i) = 25$ ) and 1 point to the synonym with the 25<sup>th</sup> rank ( $SimS(25, v_i) = 1$ ). The rank of a synonym is decreased only by false positives that are ranked higher (i.e. each of correctly identified top  $n$  synonyms has rank 0). In order to reward the top-ranked synonyms stronger the scoring function features a hyperbolic descent. For a synonym of  $v_i$  with the rank  $x$ :

$$SimS(x, v_i) = \left\{ \begin{array}{l} 0, \text{ if } x \notin synset(v_i) \\ \frac{24 * \sqrt{26}}{(\sqrt{26}-1) * \sqrt{x+1}} + 1 - \frac{24}{\sqrt{26}-1} \end{array} \right\}$$

To compare performance of different metrics the  $SimS$  values of the top 25 words in the ranked list were summed for each word of a test corpus. The total score of a similarity metric  $Sim$  is  $\sum_{i=1}^{200} \sum_{j=1}^{25} SimS(rank(RankedList(v_i, j)), v_i)$  where  $RankedList(v_i, j)$  returns the word at the position  $j$  from the ranked list produced by  $Sim$  for  $v_i$  and  $v_1, \dots, v_{200}$  are the words of the test corpus.

Besides, a combined precision and recall measure  $\Pi$  was used to evaluate the ranked lists. Given the word  $v_i$ , we examined the first  $n$  words ( $n = 1, 5, 25, 100$ ) of the ranked list returned by a similarity metric for  $v_i$  whether they belong to the  $synset(v_i)$  of the test corpus.  $\Pi(n)$  will measure precision if  $n$  is less than the size of the  $synset(v_i)$  because the maximum recall can not be reached for such  $n$  and recall otherwise because maximum precision cannot be reached for  $n > |synset(v_i)|$ . The  $\Pi$  values were averaged over 200 words. Table 1 presents the result of evaluating the similarity metrics introduced in sec. 4. The results of  $DistanceM$  confirm that regarding distance between two vertices alone is not sufficient to conclude their synonymy.  $DistanceM$  finds many related terms ranking general words with many outgoing and incoming edges higher, but it lacks the features providing the particular evidence of synonymy.  $NaivePropM$  is clearly outperformed by the both weighted metrics. The improvement relative to the  $DistanceM$  and acceptable precision of the top-ranked synonyms  $\Pi(1)$  show that considering shared properties is an adequate approach to recognition of synonyms. Ignoring the strength of semantic relation indicated by the graph and the quality of properties is the reason for the big gap in the total score and recall value ( $\Pi(100)$ ). Both weighted metrics achieved results comparable with those reported by Curran and Moens in (Curran and Moens, 2002) and Turney in (Turney, 2001). Best results of  $FirstCompM$  confirm that the criteria identified in sec. 4 such as generality of a property, abstraction from the absolute word frequency etc. are relevant for identification of synonyms.  $FirstCompM$  performed particularly better in finding synonyms with the low frequency of occur-

rence.

In another set of experiments we investigated the influence of the size of the text corpus (cf. fig. 3). The plausible assumption is the more texts are processed, the better the semantic connections between terms are reflected by the graph, the more promising results are expected. The fact that the number of vertices does not grow proportionally to the size of text corpus can be explained by word recurrence and growing filtering threshold  $\theta$ . However, the number of edges increases linearly and reflects the improving semantic coverage. As expected, every metric performs considerably better on bigger graphs. While  $NaivePropM$  seems to converge after three volumes, the both weighted metrics behave strictly monotonically increasing. Hence an improvement of results can be expected on bigger corpora. On the small text corpora the results of single metrics do not differ significantly since there is not sufficient semantic information captured by the graph, i.e. the edge and path lengths do not fully reflect the semantic relations between the words. The scores of both weighted metrics grow, though, much faster than that of  $NaivePropM$ .  $FirstCompM$  achieves the highest gradient demonstrating the biggest potential of leveraging the growing graph for finding synonymy.

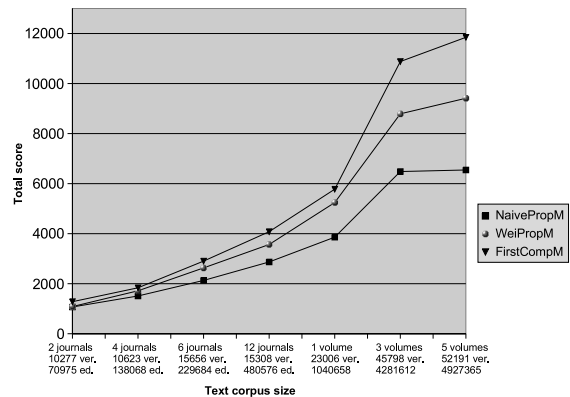


Figure 3: Influence of the size of the text corpus.

To examine the influence of the word categories results on the subsets of the text corpus corresponding to a category are compared. All metrics show similar behavior, therefore we restrict the analysis to the  $\Pi$  values of

Metric	Score	$\Pi(1)$	$\Pi(5)$	$\Pi(25)$	$\Pi(100)$
<i>DistanceM</i>	2990.7	0.20	0.208	0.199	0.38
<i>NaivePropM</i>	6546.3	0.415	0.252	0.271	0.440
<i>WeiPropM</i>	9411.7	0.54	0.351	0.398	0.607
<i>FirstCompM</i>	11848	0.575	0.412	0.472	0.637

Table 1: Results of different metrics on the test corpus

*FirstCompM* (fig. 4). Synonyms of domain-specific words are recognized better than those of abstract and everyday words. Their semantics are better reflected by the technically oriented texts. The  $\Pi$  values for abstract and everyday words are pretty similar except for the high precision of top-ranked abstract synonyms. Everyday words suffer from the fact that their properties are often too general to uniquely characterize them, which involves loss of precision. Abstract words can be extremely polysemous and have many subtle aspects that are not sufficiently covered by the texts of computer journals.

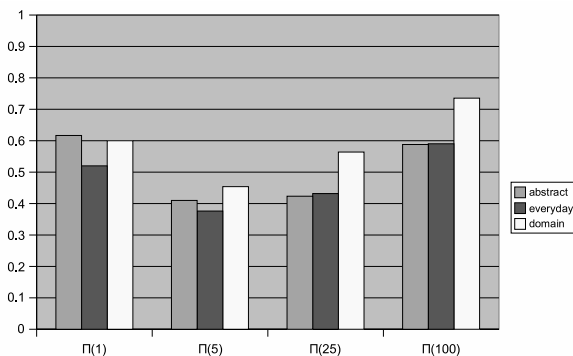


Figure 4: Dependency of  $\Pi(n)$  on word category (results of *FirstCompM* metric)

To test whether the metrics perform better for the more frequent words the test set was divided in 9 disjunctive frequency clusters (table 2). *FirstCompM* achieved considerably better results for very frequently occurring words ( $\geq 4000$  occurrences). This confirms indirectly the better results on the bigger text corpora: while low frequency does not exclude random influence, frequent occurrence involves adequate capturing of the word semantics in the graph by inserting and adjusting all relevant property edges. These results do not contradict the conclusion that *FirstCompM* is not biased towards words with a certain frequency because the mentioned bias

pertains to retrieval of synonyms with a certain frequency, whereas in this experiment the performance for different word frequencies of queried words is compared.

## 6 Conclusion

We have introduced the lexical graph as an instrument for finding semantic relations between lexical items in natural language corpora. The big advantage of the graph in comparison to other context models is that it captures not only the immediate context but establishes many transitive connections between related terms. We have verified its effectiveness searching for synonymy. Different metrics have been defined based on shortest path lengths and shared properties. Similarity metric *FirstCompM* that best leverages the graph structure achieved the best results confirming the significant role of number of shared properties, frequency of their co-occurrence and the degree of their generality for detecting of synonymy. Significantly improving results for bigger text corpora and more frequently occurring words are encouraging and promising for detection of other semantic relations. New methods that increasingly employ the graph structure e.g. regarding the lengths and number of short paths between two terms or extending the witness concept to other morphological types are the subject of further research.

## Acknowledgements

I would like to thank Heiko Kahmann for the valuable assistance in implementation and evaluation of the approach. This research is supported by NaFöG scholarship of the federal state Berlin.

## References

Vincent D. Blondel, Anah Gajardo, Maureen Heymans, Pierre Senellart, and Paul Van Dooren.

Frequency	9-249	250-499	500-999	1000-1499	1500-2499	2500-3999	4000-5499	5500-7499	>7500
Words/cluster	27	25	44	30	27	15	11	8	13
Aver. score	53.23	51.52	45.80	60.75	56.51	58.75	97.21	106.11	73.85
$\Pi(1)$	0.556	0.52	0.432	0.567	0.667	0.667	0.818	0.75	0.615
$\Pi(5)$	0.381	0.392	0.342	0.395	0.393	0.413	0.600	0.675	0.503
$\Pi(25)$	0.447	0.432	0.446	0.494	0.474	0.419	0.531	0.550	0.600
$\Pi(100)$	0.561	0.645	0.618	0.610	0.690	0.623	0.705	0.642	0.748

Table 2: Influence of word frequency on the results of *FirstCompM* metric

2004. A measure of similarity between graph vertices. With applications to synonym extraction and web searching. In *SIAM Review*, pages 647–666.
- Hsinchun Chen and Kevin J. Lynch. 1992. Automatic construction of networks of concepts characterizing document databases. In *IEEE Transactions on Systems, Man and Cybernetics*, volume 22(5), pages 885–902.
- James R. Curran and Marc Moens. 2002. Improvements in automatic thesaurus extraction. In *Proceedings of the Workshop of the ACL Special Interest Group on the Lexicon (SIGLEX)*, pages 59–66. Association for Computational Linguistics.
- M. A. Hearst. 1998. Automated discovery of Wordnet relations. In C. Fellbaum, editor, *Wordnet An Electronic Lexical Database*, pages 131–151. MIT Press, Cambridge, MA.
- Mathieu Lafourcade and Violaine Prince. 2001. Relative synonymy and conceptual vectors. In *Proceedings of the NLPRS*, Tokyo, Japan.
- Mathieu Lafourcade. 2001. Lexical sorting and lexical transfer by conceptual vectors. In *Proceedings of the First International Workshop on MultiMedia Annotation*, Tokyo, Japan.
- Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 296–304, Madison, WI.
- Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA 2000.
- OpenThesaurus. 2005. OpenThesaurus - Deutscher Thesaurus. <http://www.openthesaurus.de>.
- William Phillips and Ellen Riloff. 2002. Exploiting strong syntactic heuristics and co-training to learn semantic lexicons. In *Proceedings of the 2002 Conference on Empirical Methods in NLP*.
- Gerda Ruge. 1997. Automatic detection of thesaurus relations for information retrieval applications. *Foundations of Computer Science: Potential - Theory - Cognition*, LNCS 1337:499–506.
- Pierre P. Senellart and Vincent D. Blondel. 2003. Automatic discovery of similar words. In Michael Berry, editor, *Survey of Text Mining. Clustering, classification, and retrieval*, pages 25–44. Springer Verlag, Berlin.
- TreeTagger. 2004. <http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/>.
- Peter D. Turney. 2001. Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. *Lecture Notes in Computer Science*, 2167:491–502.
- Julie Weeds, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity.
- WordNet. 2005. <http://wordnet.princeton.edu/w3wn.html>.