Thesaurus Extension using Web Search Engines

Robert Meusel, Mathias Niepert, Kai Eckert, and Heiner Stuckenschmidt
[firstname]@informatik.uni-mannheim.de

KR & KM Research Group
University of Mannheim
Germany

Abstract. Maintaining and extending large thesauri is an important challenge facing digital libraries and IT businesses alike. In this paper we describe a method building on and extending existing methods from the areas of thesaurus maintenance, natural language processing, and machine learning to (a) extract a set of novel candidate concepts from text corpora and (b) to generate a small ranked list of suggestions for the position of these concept in an existing thesaurus. Based on a modification of the standard *tf-idf* term weighting we extract relevant concept candidates from a document corpus. We then apply a pattern-based machine learning approach on content extracted from web search engine snippets to determine the type of relation between the candidate terms and existing thesaurus concepts. The approach is evaluated with a large-scale experiment using the MeSH and WordNet thesauri as testbed.

1 Introduction

The use of thesauri in the area of document indexing and retrieval is a common approach to improve the quality of search results. Due to the fast growing number of novel concepts, manual maintenance of comprehensive thesauri is no longer feasible. A manual process would not be able to keep up with new topics that arise as a reaction to current events in the real world, quickly making their way into publications. The recent past has provided us with a number of examples, two of which we want to mention here as motivation for our contribution. In economics, the financial crisis has led to a discussion of structured financial products and terms such as "CDO" (credit debt obligation) frequently occur in documents covering current events. Nevertheless, the very same term is not included in the leading German thesaurus on business and economics. In the area of medicine, the outbreak of the H1N1 pandemic has recently sparked numerous media and research reports about the so-called "swine flu." At that point the term "swine flu" was not included in any of the major medical thesauri because it was only recently coined by the media. The current version of the MeSH thesaurus lists the term "Swine-Origin Influenza A H1N1 Virus" as a synonym for "Influenza A Virus, H1N1 Subtype" but not the more commonly used term "swine flu." In this paper we describe a possible approach to the problem of identifying important terms in text documents and semi-automatically extending

thesauri with novel concepts. The proposed system consists of three basic parts each of which we will briefly motivate by means of the swine flu example.

1. In a first step, we identify candidate terms to be included in the thesaurus. In our example this is the case for swine flu as many existing documents discuss the different aspects of swine flu, including its origin, treatment, and impact on the economy.
2. Once we decide that the term "swine flu" should be included in the thesaurus, we have to identify a location that is most appropriate. This step requires a deeper understanding of the concept "swine flu" since we want to place it in the disease branch and not the animal branch of the thesaurus. In particular, the term should be classified next to the concept "Influenza A Virus, H1N1 Subtype."
3. After deciding to place "swine flu" close to 'Influenza A Virus, H1N1 Subtype' one still needs to determine the relation of the two concepts. In particular, we have to decide whether the new term should be regarded as a synonym or whether it should be included as a concept of its own - either as hyponym or hypernym or whether the similarity of the two terms was incidental.

The contributions of this paper are the following: (1) We propose methods for carrying out the three steps mentioned above by looking at the literature and adapting existing approaches. (2) We present a large-scale experiment applying these methods to extend parts of the MeSH thesaurus with new terms extracted from documents. (3) We present detailed results on the use of web search engines as a means for generating feature sets for learning the correct relation of new and existing terms in step (3).

The paper is structured as followed: In section 2 we explain where our work has to be classified and what other researchers have accomplished in this area. Section 3 includes the detailed description of our approach and the necessary foundations. The experiments and their evaluation and results are summarized in Section 4. In the conclusion (section 5) we summarize the individual results and present a short outlook for future research in the area.

2 Related Work

Nguyen et al. [13] used lexico-syntactic patterns mined from the online encyclopedia *wikipedia.org* to extract relations between terms. Gillam et al. [5] describe a combination of term extraction, co-occurrence-based measures and predefined linguistic patterns to construct a thesaurus structure from domain-specific collections of texts. Another combination of these techniques using hidden markov random fields is presented by Kaji and Kitsuregawa [9]. Witschel [17] employs a decision tree algorithm to insert novel concepts into a taxonomy. Kermanidis et al. [10] present with Eksairesis a system for ontology building from unstructured text adaptable to different domains and languages. For the process of term extraction they use two corpora, a balanced corpus and a domain-specific corpus.

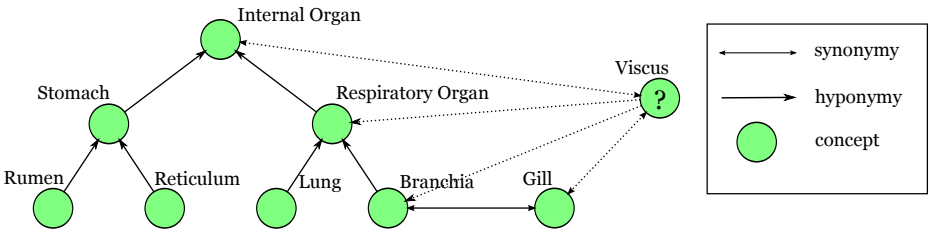

Fig. 1. Fragment of the WordNet thesaurus.

The semantic relations are learned from syntactic schemata, an approach that is applicable to corpora written in languages without strict sentence word ordering such as modern Greek. Niepert, Buckner and Allen[14] combine statistical NLP methods with expert feedback and logic programming to extend a philosophy thesaurus. This approach is combined with crowdsourcing strategies in Eckert et al.[4].

Many methods focus only on the extraction of synonyms from text corpora: Turney [15] computes the similarity between synonym candidates leveraging the number of hits returned for different combinations of search terms. Matsuo et al. [12] apply co-occurrence measures on search engine results to cluster words. Curran [3] combines several methods for synonym extraction and shows that the combination outperforms each of the single methods, including Grefenstette's approach [6]. In some cases, special resources such as bilingual corpora or dictionaries are available to support specialized methods for automatic thesaurus construction. Wu and Zhou [18] describe a combination of such methods to extract synonyms. Other techniques using multilingual corpora are described by van der Plas and Tiedemann [16] and Kageura et al. [8].

3 Method Description

Let us assume we are given a thesaurus $\mathcal{T}$ that needs to be extended with novel concepts. The process of thesaurus extension can be divided in two major phases. First, concept candidates have to be extracted from document collections and other textual content. In order to achieve satisfiable results it is necessary that the text corpora under consideration are semantically related to the concepts in the thesaurus. For instance, if we want to extend a thesaurus of medical terms we would have to choose a document collection covering medical topics. Given a set of candidate terms, the second step of thesaurus extension involves the classification of these candidates as either synonyms or hyponyms of already existing thesaurus concepts.

Figure 1 depicts a typical instance of the thesaurus extension problem. We propose a method supporting the knowledge modeler during both of these phases by (a) extracting terms from text corpora using a novel extraction method based on the well-known *tf-idf* measure and (b) by generating, for each of the extracted

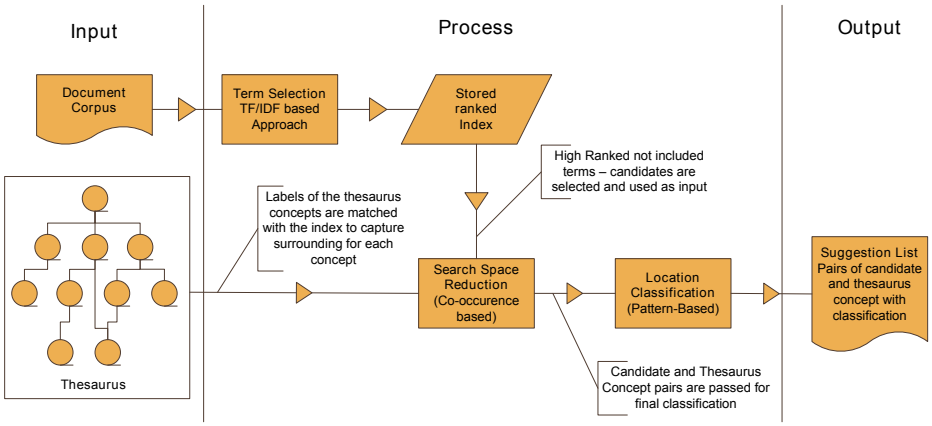

Fig. 2. The workflow of the thesaurus extension system.

concept candidates, a reasonable sized set of suggestions for its position in the thesaurus. For the latter, we distinguish between synonymy and hyponymy relationships. Figure 2 depicts a work-flow of the proposed thesauri extension support system. In the remainder of this section we describe the two components of the system in more detail.

3.1 Term Selection

Term selection is the process of extracting terms that could serve as concepts in the thesaurus. This is usually done by applying statistical co-occurrence measures to a corpus of text documents. In order to quantify the importance of a term t in a corpus D we first compute the *tf-idf* value $w_{t,d}$ of term t in document d. We found that applying the *tf-idf* variant with (a) logarithmic term frequency weighting, (b) logarithmic document frequency weighting, and (c) cosine normalization yielded the best results. More formally, we computed the cosine normalized *tf-idf* value $w_{t,d}^{norm}$ for each term t and each document d according to Equation 1.

$$w_{t,d}^{norm} = \frac{w_{t,d}}{\sqrt{\sum_{t' \in d} w_{t',d}}} \quad \text{with} \quad w_{t,d} = (1 + \log(tf_{t,d})) * \log \frac{|D|}{df_t} \tag{1}$$

Since we want to assess the importance of a term t not only for a single document but the entire corpus, we compute the mean $\overline{w}_t$ of the *tf-idf* values over all documents in which term t occurs at least once.

$$\overline{w}_t = \frac{\sum_{d \in D} w_{t,d}^{norm}}{df_t} \tag{2}$$

We finally assign the importance weight $\hat{w}_t$ to term t by multiplying the squared value $\overline{w}_t$ with the logarithm of the document frequency df_t.

$$\hat{w}_t = \log(df_t + 1) \times \overline{w}_t^2 \tag{3}$$

The intuition behind this approach is that terms that occur in more documents are more likely to be concept candidates for a thesaurus covering these documents. The presented importance measure $\hat{w}_t$, therefore, combines the average importance of a term relative to each document in the corpus with the importance of the term relative to the entire corpus.

3.2 Pattern-Based Position Extraction

In a second step, the previously extracted concept candidates are *classified* in the existing thesaurus. Classification is the process of finding concepts in the thesaurus that are potential hypernyms and synonyms, respectively, for each of the candidate concepts. This process is also often referred to as *position extraction*. We apply established machine learning approaches to learn lexico-syntactic patterns from search engine results. Typical patterns for concepts C_1 and C_2 are, for instance, [C_1 *is a* C_2] for hyponymy and [C_1 *is also a* C_2] for synonymy relationships. Instead of only using a predefined set of patterns [7], however, we learn these patterns from text snippets of search engines [2] using existing thesauri as training data. The learned patterns are then used as features for the classification of the relationship between each concept candidate and existing thesaurus concepts. Since we are mainly interested in hyponymy and synonymy relationships, we need to train at least two different binary classifiers. Fortunately, the classifiers can be trained with concepts pairs contained in existing domain thesauri.

The pattern extraction approach of the proposed system is based on the method presented by Bollegala et al. [2]. Instead of retrieving lexico-syntactic patterns to asses the semantic similarity of term pairs, however, we extract the patterns also for the purpose of classifying relationship types as either synonymy or hyponymy. For each pair of concepts (C_1, C_2) of which we know the relationship because it is contained in a training thesaurus, we send the query "C_1" +"C_2" to a web search engine. The returned text snippet is processed to extract all n-grams ($2 \leq n \leq 6$) that match the pattern "$C_1 X^* C_2$", where X can be any combination of up to four space-separated word or punctuation tokens. For instance, assume the training thesaurus contains the concepts "car" and "vehicle" with car being a hyponym of vehicle. The method would query a search engine with the string "car" +"vehicle". Let us assume that one of the returned text snippet is "every car is a vehicle." In this case, the method would extract the pattern "car is a vehicle". This pattern would be added to the list of potential hyponymy patterns with "car" and "vehicle" substituted with matching placeholders. Of course, the set of patterns extracted this way is too large to be used directly for machine learning algorithms. Therefore, we rank the patterns according to their ability to distinguish between the types of relationships we are interested in. For both the synonymy and hyponymy relationship we rank the extracted patterns according to the chi-square statistic [2]. For every pattern v we determine its frequency p_v in snippets for hyponymous (synonymous) word pairs and its frequency n_v in snippets for non-hyponymous (non-synonymous)

word pairs. Let P denote the total frequency of all patterns in snippets for hyponymous (synonymous) word pairs and N the total frequency of all patterns in snippets for non-hyponymous (non-synonymous) word pairs. We calculate the chi-square value (Bollegala et al. [2]) for each pattern as follows:

$$\chi_v^2 = \frac{(P+N)(p_v(N-n_v) - n_v(P-p_v))^2}{PN(p_v+n_v)(P+N-p_v-n_v)} \tag{4}$$

From the initially extracted set of patterns we kept only the 80 highest ranked patterns extracted with WordNet as training thesaurus and the 60 highest ranked patterns with the medical subject headings (MeSH) thesaurus as training thesaurus. The feature vector for the machine learning algorithms consists of the normalized frequencies for these top-ranked patterns. Finally, we learn a support vector machine with linear kernel, a support vector machine with radial basis function (RBF) kernel, and a decision tree algorithm (J48) using the generated feature vectors. Figure 1 depicts a typical instance of the thesaurus extension problem. The concept candidate "Viscus", which has been extracted from a text corpus, needs to be classified in the existing thesaurus. The thesaurus extension support system provides, for each candidate concept, a small ranked list of potential positions in the thesaurus. In the following section we report on the empirical evaluation of the presented approach.

4 Experimental Evaluation

Most thesauri are comprised of a large number of concepts and, for every candidate concept, we would have to send a query to a web search engine for every of the thesaurus' concepts. Hence, we have to reduce the amount of potential thesaurus positions for any given candidate concept. To achieve such a search space reduction we compute, for every candidate concept that needs to be classified, its similarity to each of the thesaurus concepts using the weighted Jaccard value of its surrounding words (Lin [11]). Then, for each concept candidate, only the top-k most similar thesaurus concepts are considered for the pattern based approach. In the following we call the concepts which are included in the top-k set the *similar concepts*. The thesaurus concepts that share a hyponymy or synonymy relation with a candidate concept are referred to as *positional concepts*.

While the pattern extraction approach would work with any search engine, we decided to use the Yahoo search engine API[1] as it is less restrictive on the allowed number of queries per day. A single query with the API took up to three seconds. To evaluate and test our methods we used a thesauri extracted from the MeSH thesaurus of the year 2008[2]. The thesaurus was created by combining all concepts located under the top-level concept *anatomy* (1611 concepts) with all concept located under the top-level concept *humanity* (186 concepts). For each concept in these thesauri we retrieved the most relevant documents from

[1] http://developer.yahoo.com/
[2] http://www.nlm.nih.gov/mesh/

Training data	Classification task	SVM (lin)	SVM (RBF)	Decision tree
WordNet	synonym vs. no synonym	86 %	54 %	98 %
WordNet	hyponym vs. no hyponym	73 %	63 %	82 %
WordNet	synonym vs. hyponym	70 %	50 %	71 %
WordNet	synonym vs. hyponym vs. none	58 %	47 %	70 %
MeSH	synonym vs. no synonym	71 %	59 %	85 %
MeSH	hyponym vs. no hyponym	74 %	60 %	87 %
MeSH	synonym vs. no synonym	53 %	52 %	68 %
MeSH	synonym vs. hyponym vs. none	51 %	40 %	68 %

Table 1. Accuracy results of the three machine learning approaches and two thesauri for different classification tasks.

PubMed[3] of the years between 2005 and 2008. The final document corpus included 13392 documents for the *anatomy* thesaurus and 1468 documents for the *humanity* thesaurus. We chose WordNet 3.0 as a second thesaurus for the experiments, primarily since this allows us to compare the results to those reported in Bollegala et al. [1].

For each of the three classes "synonymy", "hyponymy", and "neither synonymy nor hyponymy" we sampled 300 pairs of concepts belonging to the respective class. For the MeSH training set, these pairs were randomly sampled from the MeSH thesaurus excluding the previously constructed anatomy/humanity sub-thesaurus. Similarly, to create the WordNet training set, we randomly sampled 300 negative and positive training pairs for each class from WordNet. For testing, we isolated 100 concepts each from the anatomy/humanity sub-thesaurus and from WordNet. These concepts serve as candidate concepts and the goal is to evaluate whether our approach can identify their correct positions. For both the 100 MeSH and WordNet candidate concepts we determined the top 100 most similar concepts in the MeSH and WordNet thesaurus, respectively, by applying the above-mentioned co-occurrence similarity measure. On average, 97 percent of the correct positions for each candidate concept were included in this set for WordNet and 90 percent for the MeSH thesaurus. This indicates that the Jaccard similarity measure is able to exclude the majority of all concept positions while retaining most of the correct positional concepts.

For each of the 100 concept candidates, we applied the trained classifiers on the set of the previously ranked 100 most similar concepts, resulting in 10000 classifications instances for each combination of thesaurus (MeSH or WordNet), classifier (linear SVM, RBF SVN, decision tree), and classification task. The accuracy values ((true positives + true negatives) / all instances) of these experiments are shown in Table 1. Evidently, the accuracy of the classifiers is strongly influenced by the properties of the thesauri. For instance, for the synonymy classification task, we achieved an accuracy of 86 percent with a linear SVM for WordNet but only an accuracy of 71 percent for the MeSH thesaurus. Not surprisingly, the three-class classification problem is more difficult and the

[3] http://www.ncbi.nlm.nih.gov/pubmed/

Training data	Classification task	SVM (linear)	Decision tree
WordNet	synonym vs. no synonym	7.8 %	6.7 %
WordNet	hyponym vs. no hyponym	10.6 %	15.4 %
MeSH	synonym vs. no synonym	3.6 %	12.7 %
MeSH	hyponym vs. no hyponym	6.1 %	14.1 %

Table 2. Percentage of candidate concepts wrongly classified as synonyms (hyponyms) by the linear support vector machine (SVM) and the decision tree algorithm.

approach is not as accurate as for the binary classification tasks. An additional observation is that the classification results for the hyponymy vs. synonymy problem are rather poor pointing to the semantic similarity of the synonymy and hyponymy relations.

Furthermore, the results reveal that the decision tree algorithm (J48) leads to more accurate classification results for the majority of the tasks. The accuracy of the J48 classifier is on average 11.6 percent better than the linear SVM classifier and 24.1 percent more accurate than the radial basis function SVM. This is especially interesting because pattern based machine learning approaches mostly employ support vector machines for classification. Proper parameter tuning could close the performance gap between the two approaches, however, this is often not possible in real-world applications. While the decision tree approach is superior in accuracy the linear SVM classifier is more precise. Table 2 shows the percentage of false positives for the synonymy and hyponymy classes for both the MeSH and WordNet thesaurus. Except for the synonymy vs. no synonymy classification problem the linear SVM algorithm results in fewer false positives. A thesaurus maintenance system should support the knowledge modeler by reducing the amount of novel concept/position pairs without excluding correct ones. Therefore, we are especially interested in a high recall and moderate precision making the decision tree algorithm the preferred choice for the thesaurus maintenance setting.

For a librarian or knowledge modeler, the main application of the support system is to locate the correct position of the candidate concepts in the thesaurus. Let us assume we are given the concept candidate “tummy” and that we need to determine its position in the thesaurus fragment depicted in Figure 1. Now, two pieces of information will lead us to the correct location. The first one being that “tummy” is a hyponym of “internal organ” and the second being that “tummy” is a synonym of “stomach.” In an additional experiment we evaluated, for each concept candidate, in how many cases we were able to determine the correct position in the target thesaurus. Hence, for each concept candidate, we looked at the set of concepts in the thesaurus which the pattern-based approach classified as either synonyms or hyponyms and checked whether at least on of these concepts led us to the correct position. The size of this set was 14 on average, meaning that, on average, the number of choices was reduced from 100 to 14. Figure 3 lists the percentage of cases for which we could determine the correct position for the MeSH thesaurus. We also widened the graph distance

Graph distance	MeSH thesaurus
1	85%
2	95%
3	99%
4	100%

Table 3. Fraction of candidate concepts for which the correct position in the thesaurus could be inferred using the pattern-based classification results; and considering a graph distance of $1 \leq n \leq 4$.

to the correct position from 1 to 4, where the graph distance 1 represents direct synonymy or hyponymy relations. The suggested position was at most 4 edges away from the correct one.

5 Conclusion and Future Work

The results of the experimental evaluation demonstrate that the presented approach has the potential to support and speed-up the laborious task of thesaurus construction and maintenance. The concept candidate ranking based on the adapted *tf-idf* relevance measure (see Equation 3) could identify most of the significant terms of a text corpus. The combination of co-occurrence guided search space reduction and pattern-based position extraction results in accurate classification results, leaving a drastically reduced number of choices to the knowledge modeler. Furthermore, the experiments indicate that web search engine snippets contain enough information to also learn lexico-syntactic patterns for the problem of hyponymy extraction. The combination of synonymy and hyponymy classification allows us to locate, for each extracted candidate concept, the appropriate position in the thesaurus. We believe only slight modifications are necessary to adapt the system to several important real-world use cases including thesaurus maintenance for digital libraries and information retrieval systems. Both of these use cases are important to businesses as well as university libraries.

We intend to conduct more experiments on different heterogeneous thesauri, attempting to relate thesaurus properties to the performance of the pattern based approach. Based on these finding we hope to be able to tune the machine learning approach to achieve improved accuracy and performance, making the approach more suitable for domain-specific and large-scale thesauri. Furthermore, instead of merely extending a thesaurus, we will try to adapt the approach to construct thesauri entirely from scratch using only text corpora and web search engines. A bottleneck of the pattern based approach is the time it takes to query the web search engine. In this work, we reduced the number of pairs by using a co-occurrence similarity measure. In future work, however, we will investigate additional methods to reduce the number of concept positions that have to be visited in the thesaurus. For instance, having strong evidence that a candidate concept is not a hyponym of a thesaurus concept C we can immediately infer that it can also not be a hyponymy of any of C's descendants. This would allow

us to prune entire sub-trees in the thesaurus, drastically reducing the number of pairs that have to be send to the web search engine. Another idea is to not only apply shallow parsing strategies to extract lexical pattern but also more sophisticated approaches such as POS tagging and deep syntax parsing.

References

1. D. Bollegala, Y. Matsuo, and M. Ishizuka. An integrated approach to measuring semantic similarity between words using information available on the web. In *Proceedings of HLT-NAACL*, pages 340–347, 2007.
2. D. Bollegala, Y. Matsuo, and M. Ishizuka. Measuring semantic similarity between words using web search engines. In *Proceedings of WWW*, pages 757–766, 2007.
3. J. R. Curran. Ensemble methods for automatic thesaurus extraction. In *Proceedings of ACL*, pages 222–229, 2002.
4. K. Eckert, M. Niepert, C. Niemann, C. Buckner, C. Allen, and H. Stuckenschmidt. Crowdsourcing the assembly of concept hierarchies. In *Proceedings of JCDL*, 2010.
5. L. Gillam, M. Tariq, and K. Ahmad. Terminology and the construction of ontology. *Terminology*, 11:55–81, 2005.
6. G. Grefenstette. *Explorations in Automatic Thesaurus Discovery*. Springer, Heidelberg, 1994.
7. M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics, Nantes, France*, 1992.
8. K. Kageura, K. Tsuji, and A. N. Aizawa. Automatic thesaurus generation through multiple filtering. In *In Proceedings COLING*, pages 397–403, 2000.
9. N. Kaji and M. Kitsuregawa. Using hidden markov random fields to combine distributional and pattern-based word clustering. In *Proceedings of COLING*, pages 401–408, 2008.
10. M. M. Katia Lida Kermanidis, Aristomenis Thanopoulos and N. Fakotakis. Eksairesis: A domain-adaptable system for ontology building from unstructured text. In *Proceedings of LREC*, 2008.
11. D. Lin. An information-theoretic definition of similarity. In *In Proceedings of ICML*, pages 296–304, 1998.
12. Y. Matsuo, T. Sakaki, K. Uchiyama, and M. Ishizuka. Graph-based word clustering using a web search engine. In *Proceedings of EMNLP*, pages 542–550, 2006.
13. D. P. T. Nguyen, Y. Matsuo, and M. Ishizuka. Exploiting syntactic and semantic information for relation extraction from wikipedia. In *Proceedings of IJCAI*, 2007.
14. M. Niepert, C. Buckner, and C. Allen. A dynamic ontology for a dynamic reference work. In *Proceedings of JCDL*, pages 288–297, 2007.
15. P. D. Turney. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *Proceedings of EMCL*, pages 491–502, 2001.
16. L. van der Plas and J. Tiedemann. Finding synonyms using automatic word alignment and measures of distributional similarity. In *Proceedings of COLING*, pages 866–873, 2006.
17. H. F. Witschel. Using decision trees and text mining techniques for extending taxonomies. In *Proceedings of the Workshop on Learning and Extending Lexical Ontologies by Using Machine Learning Methods*, 2005.
18. H. Wu and M. Zhou. Optimizing synonym extraction using monolingual and bilingual resources. In *Proceedings of the second international workshop on Paraphrasing*, pages 72–79, 2003.