Integrating Pattern-based and Distributional Similarity Methods for Lexical Entailment Acquisition

Shachar Mirkin
School of Computer Science and Engineering
The Hebrew University, Jerusalem, Israel, 91904
`mirkin@cs.huji.ac.il`

Ido Dagan **Maayan Geffet**
Department of Computer Science
Bar-Ilan University, Ramat Gan, Israel, 52900
`{dagan,zitima}@cs.biu.ac.il`

Abstract

This paper addresses the problem of acquiring lexical semantic relationships, applied to the lexical entailment relation. Our main contribution is a novel conceptual integration between the two distinct acquisition paradigms for lexical relations – the pattern-based and the distributional similarity approaches. The integrated method exploits mutual complementary information of the two approaches to obtain candidate relations and informative characterizing features. Then, a small size training set is used to construct a more accurate supervised classifier, showing significant increase in both recall and precision over the original approaches.

1 Introduction

Learning lexical semantic relationships is a fundamental task needed for most text understanding applications. Several types of lexical semantic relations were proposed as a goal for automatic acquisition. These include lexical ontological relations such as synonymy, hyponymy and meronymy, aiming to automate the construction of WordNet-style relations. Another common target is learning general distributional similarity between words, following Harris' Distributional Hypothesis (Harris, 1968). Recently, an applied notion of entailment between lexical items was proposed as capturing major inference needs which cut across multiple semantic relationship types (see Section 2 for further background).

The literature suggests two major approaches for learning lexical semantic relations: distributional similarity and pattern-based. The first approach recognizes that two words (or two multi-word terms) are semantically similar based on distributional similarity of the different contexts in which the two words occur. The distributional method identifies a somewhat loose notion of semantic similarity, such as between *company* and *government*, which does not ensure that the meaning of one word can be substituted by the other. The second approach is based on identifying joint occurrences of the two words within particular patterns, which typically indicate directly concrete semantic relationships. The pattern-based approach tends to yield more accurate hyponymy and (some) meronymy relations, but is less suited to acquire synonyms which only rarely co-occur within short patterns in texts. It should be noted that the pattern-based approach is commonly applied also for information and knowledge extraction to acquire factual instances of concrete meaning relationships (e.g. *born in*, *located at*) rather than generic lexical semantic relationships in the language.

While the two acquisition approaches are largely complementary, there have been just few attempts to combine them, usually by pipeline architecture. In this paper we propose a methodology for integrating distributional similarity with the pattern-based approach. In particular, we focus on learning the lexical entailment relationship between common nouns and noun phrases (to be distinguished from learning relationships for proper nouns, which usually falls within the knowledge acquisition paradigm).

The underlying idea is to first identify candidate relationships by both the distributional approach, which is applied exhaustively to a local corpus, and the pattern-based approach, applied to the web. Next, each candidate is represented by a unified set of distributional and pattern-based features. Finally, using a small training set we devise a supervised (SVM) model that classifies new candidate relations as correct or incorrect.

To implement the integrated approach we developed state of the art pattern-based acquisition methods and utilized a distributional similarity method that was previously shown to provide superior performance for lexical entailment acquisition. Our empirical results show that the integrated method significantly outperforms each approach in isolation, as well as the naïve combination of their outputs. Overall, our method reveals complementary types of information that can be obtained from the two approaches.

2 Background

2.1 Distributional Similarity and Lexical Entailment

The general idea behind distributional similarity is that words which occur within similar contexts are semantically similar (Harris, 1968). In a computational framework, words are represented by feature vectors, where features are context words weighted by a function of their statistical association with the target word. The degree of similarity between two target words is then determined by a vector comparison function. Amongst the many proposals for distributional similarity measures, (Lin, 1998) is maybe the most widely used one, while (Weeds et al., 2004) provides a typical example for recent research. Distributional similarity measures are typically computed through exhaustive processing of a corpus, and are therefore applicable to corpora of bounded size.

It was noted recently by Geffet and Dagan (2004, 2005) that distributional similarity captures a quite loose notion of semantic similarity, as exemplified by the pair *country – party* (identified by Lin's similarity measure). Consequently, they proposed a definition for the *lexical entailment* relation, which conforms to the general framework of applied textual entailment (Dagan et al., 2005). Generally speaking, a word w lexically entails another word v if w can substitute v in some contexts while implying v's original meaning. It was suggested that lexical entailment captures major application needs in modeling lexical variability, generalized over several types of known ontological relationships. For example, in Question Answering (QA), the word *company* in a question can be substituted in the text by *firm* (synonym), *automaker* (hyponym) or *subsidiary* (meronym), all of which entail *company*. Typically, hyponyms entail their hypernyms and synonyms entail each other, while entailment holds for meronymy only in certain cases.

In this paper we investigate automatic acquisition of the lexical entailment relation. For the distributional similarity component we employ the similarity scheme of (Geffet and Dagan, 2004), which was shown to yield improved predictions of (non-directional) lexical entailment pairs. This scheme utilizes the symmetric similarity measure of (Lin, 1998) to induce improved feature weights via bootstrapping. These weights identify the most characteristic features of each word, yielding cleaner feature vector representations and better similarity assessments.

2.2 Pattern-based Approaches

Hearst (1992) pioneered the use of lexical-syntactic patterns for automatic extraction of lexical semantic relationships. She acquired hyponymy relations based on a small predefined set of highly indicative patterns, such as "*X, . . . , Y and/or other Z*", and "*Z such as X, . . . and/or Y*", where X and Y are extracted as hyponyms of Z. Similar techniques were further applied to predict hyponymy and meronymy relationships using lexical or lexico-syntactic patterns (Berland and Charniak, 1999; Sundblad, 2002), and web page structure was exploited to extract hyponymy relationships by Shinzato and Torisawa (2004). Chklovski and Pantel (2004) used patterns to extract a set of relations between verbs, such as similarity, strength and antonymy. Synonyms, on the other hand, are rarely found in such patterns. In addition to their use for learning lexical semantic relations, patterns were commonly used to learn instances of concrete semantic relations for Information Extraction (IE) and QA, as in (Riloff and Shepherd, 1997; Ravichandran and Hovy, 2002; Yangarber et al., 2000).

Patterns identify rather specific and informative structures within particular co-occurrences of the related words. Consequently, they are relatively reliable and tend to be more accurate than distributional evidence. On the other hand, they are susceptive to data sparseness in a limited size corpus. To obtain sufficient coverage, recent works such as (Chklovski and Pantel, 2004) applied pattern-based approaches to the web. These methods form search engine queries that match likely pattern instances, which may be verified by post-processing the retrieved texts.

Another extension of the approach was automatic enrichment of the pattern set through bootstrapping. Initially, some instances of the sought

1	NP_1 such as NP_2
2	Such NP_1 as NP_2
3	NP_1 or other NP_2
4	NP_1 and other NP_2
5	NP_1 ADV known as NP_2
6	NP_1 especially NP_2
7	NP_1 like NP_2
8	NP_1 including NP_2
9	NP_1-sg is (a OR an) NP_2-sg
10	NP_1-sg (a OR an) NP_2-sg
11	NP_1-pl are NP_2-pl

Table 1: The patterns we used for entailment acquisition based on (Hearst, 1992) and (Pantel et al., 2004). Capitalized terms indicate variables. *pl* and *sg* stand for plural and singular forms.

relation are found based on a set of manually defined patterns. Then, additional co-occurrences of the related terms are retrieved, from which new patterns are extracted (Riloff and Jones, 1999; Pantel et al., 2004). Eventually, the list of effective patterns found for ontological relations has pretty much converged in the literature. Amongst these, Table 1 lists the patterns that were utilized in our work.

Finally, the selection of candidate pairs for a target relation was usually based on some function over the statistics of matched patterns. To perform more systematic selection Etzioni et al. (2004) applied a supervised Machine Learning algorithm (Naïve Bayes), using pattern statistics as features. Their work was done within the IE framework, aiming to extract semantic relation instances for proper nouns, which occur quite frequently in indicative patterns. In our work we incorporate and extend the supervised learning step for the more difficult task of acquiring general language relationships between common nouns.

2.3 Combined Approaches

It can be noticed that the pattern-based and distributional approaches have certain complementary properties. The pattern-based method tends to be more precise, and also indicates the direction of the relationship between the candidate terms. The distributional similarity approach is more exhaustive and suitable to detect symmetric synonymy relations. Few recent attempts on related (though different) tasks were made to classify (Lin et al., 2003) and label (Pantel and Ravichandran, 2004) distributional similarity output using lexical-syntactic patterns, in a pipeline architecture. We aim to achieve tighter integration of the two approaches, as described next.

3 An Integrated Approach for Lexical Entailment Acquisition

This section describes our integrated approach for acquiring lexical entailment relationships, applied to common nouns. The algorithm receives as input a *target term* and aims to acquire a set of terms that either entail or are entailed by it. We denote a pair consisting of the input target term and an acquired entailing/entailed term as *entailment pair*. Entailment pairs are directional, as in *bank* → *company*.

Our approach applies a supervised learning scheme, using SVM, to classify candidate entailment pairs as correct or incorrect. The SVM training phase is applied to a small constant number of training pairs, yielding a classification model that is then used to classify new test entailment pairs. The designated training set is also used to tune some additional parameters of the method. Overall, the method consists of the following main components:

1: Acquiring candidate entailment pairs for the input term by pattern-based and distributional similarity methods (Section 3.2);

2: Constructing a feature set for all candidates based on pattern-based and distributional information (Section 3.3);

3: Applying SVM training and classification to the candidate pairs (Section 3.4).

The first two components, of acquiring candidate pairs and collecting features for them, utilize a generic module for pattern-based extraction from the web, which is described first in Section 3.1.

3.1 Pattern-based Extraction Module

The general pattern-based extraction module receives as input a set of lexical-syntactic patterns (as in Table 1) and either a target term or a candidate pair of terms. It then searches the web for occurrences of the patterns with the input term(s). A small set of effective queries is created for each pattern-terms combination, aiming to retrieve as much relevant data with as few queries as possible.

Each pattern has two variable slots to be instantiated by candidate terms for the sought relation. Accordingly, the extraction module can be used in two modes: (a) receiving a single target term as input and searching for instantiations of the other variable to identify candidate related terms (as in Section 3.2); (b) receiving a candidate pair of terms for the relation and searching pattern instances with both terms, in order to validate and collect information about the relationship between the terms (as in Section 3.3). Google proximity search[1] provides a useful tool for these purposes, as it allows using a wildcard which might match either an un-instantiated term or optional words such as modifiers. For example, the query *"such ** as *** (war OR wars)"* is one of the queries created for the input pattern *such* NP_1 *as* NP_2 and the input target term *war*, allowing new terms to match the first pattern variable. For the candidate entailment pair *war → struggle*, the first variable is instantiated as well. The corresponding query would be: *"such * (struggle OR struggles) as *** (war OR wars)"*. This technique allows matching terms that are sub-parts of more complex noun phrases as well as multi-word terms.

The automatically constructed queries, covering the possible combinations of multiple wildcards, are submitted to Google[2] and a specified number of snippets is downloaded, while avoiding duplicates. The snippets are passed through a word splitter and a sentence segmenter[3], while filtering individual sentences that do not contain all search terms. Next, the sentences are processed with the OpenNLP[4] POS tagger and NP chunker. Finally, pattern-specific regular expressions over the chunked sentences are applied to verify that the instantiated pattern indeed occurs in the sentence, and to identify variable instantiations.

On average, this method extracted more than 3300 relationship instances for every 1MB of downloaded text, almost third of them contained multi-word terms.

3.2 Candidate Acquisition

Given an input target term we first employ pattern-based extraction to acquire entailment pair candidates and then augment the candidate set with pairs obtained through distributional similarity.

3.2.1 Pattern-based Candidates

At the candidate acquisition phase pattern instances are searched with one input target term, looking for instantiations of the other pattern variable to become the candidate related term (the first querying mode described in Section 3.1). We construct two types of queries, in which the target term is either the first or second variable in the pattern, which corresponds to finding either entailing or entailed terms that instantiate the other variable.

In the candidate acquisition phase we utilized patterns 1-8 in Table 1, which we empirically found as most suitable for identifying directional lexical entailment pairs. Patterns 9-11 are not used at this stage as they produce too much noise when searched with only one instantiated variable. About 35 queries are created for each target term in each entailment direction for each of the 8 patterns. For every query, the first n snippets are downloaded (we used $n=50$). The downloaded snippets are processed as described in Section 3.1, and candidate related terms are extracted, yielding candidate entailment pairs with the input target term.

Quite often the entailment relation holds between multi-word noun-phrases rather than merely between their heads. For example, *trade center* lexically entails *shopping complex*, while *center* does not necessarily entail *complex*. On the other hand, many complex multi-word noun phrases are too rare to make a statistically based decision about their relation with other terms. Hence, we apply the following two criteria to balance these constraints:

1. For the entailing term we extract only the complete noun-chunk which instantiate the pattern. For example: we extract *housing project → complex*, but do not extract *project* as entailing *complex* since the head noun alone is often too general to entail the other term.
2. For the entailed term we extract both the complete noun-phrase and its head in order to create two separate candidate entailment pairs with the entailing term, which will be judged eventually according to their overall statistics.

As it turns out, a large portion of the extracted pairs constitute trivial hyponymy relations, where one term is a modified version of the other, like *low interest loan → loan*. These pairs were removed, along with numerous pairs including proper nouns, following the goal of learning en-

[1] Previously used by (Chklovski and Pantel, 2004).
[2] http://www.google.com/apis/
[3] Available from the University of Illinois at Urbana-Champaign, http://l2r.cs.uiuc.edu/~cogcomp/tools.php
[4] www.opennlp.sourceforge.net/

tailment relationships for distinct common nouns.

Finally, we filter out the candidate pairs whose frequency in the extracted patterns is less than a threshold, which was set empirically to 3. Using a lower threshold yielded poor precision, while a threshold of 4 decreased recall substantially with just little effect on precision.

3.2.2 Distributional Similarity Candidates

As mentioned in Section 2, we employ the distributional similarity measure of (Geffet and Dagan, 2004) (denoted here *GD04* for brevity), which was found effective for extracting non-directional lexical entailment pairs. Using local corpus statistics, this algorithm produces for each target noun a scored list of up to a few hundred words with positive distributional similarity scores.

Next we need to determine an optimal threshold for the similarity score, considering words above it as likely entailment candidates. To tune such a threshold we followed the original methodology used to evaluate *GD04*. First, the top-*k* (*k=40*) similarities of each training term are manually annotated by the lexical entailment criterion (see Section 4.1). Then, the similarity value which yields the maximal micro-averaged F1 score is selected as threshold, suggesting an optimal recall-precision tradeoff. The selected threshold is then used to filter the candidate similarity lists of the test words.

Finally, we remove all entailment pairs that already appear in the candidate set of the pattern-based approach, in either direction (recall that the distributional candidates are non-directional). Each of the remaining candidates generates two directional pairs which are added to the unified candidate set of the two approaches.

3.3 Feature Construction

Next, each candidate is represented by a set of features, suitable for supervised classification. To this end we developed a novel feature set based on both pattern-based and distributional data.

To obtain pattern statistics for each pair, the second mode of the pattern-based extraction module is applied (see Section 3.1). As in this case, both variables in the pattern are instantiated by the terms of the pair, we could use all eleven patterns in Table 1, creating a total of about 55 queries per pair and downloading *m*=20 snippets for each query. The downloaded snippets are processed as described in Section 3.1 to identify pattern matches and obtain relevant statistics for feature scores.

Following is the list of feature types computed for each candidate pair. The feature set was designed specifically for the task of extracting the complementary information of the two methods.

Conditional Pattern Probability**:** This type of feature is created for each of the 11 individual patterns. The feature value is the estimated conditional probability of having the pattern matched in a sentence given that the pair of terms does appear in the sentence (calculated as the fraction of pattern matches for the pair amongst all unique sentences that contain the pair). This feature yields normalized scores for pattern matches regardless of the number of snippets retrieved for the given pair. This normalization is important in order to bring to equal grounds candidate pairs identified through either the pattern-based or distributional approaches, since the latter tend to occur less frequently in patterns.

Aggregated Conditional Pattern Probability: This single feature is the conditional probability that *any* of the patterns match in a retrieved sentence, given that the two terms appear in it. It is calculated like the previous feature, with counts aggregated over all patterns, and aims to capture overall appearance of the pair in patterns, regardless of the specific pattern.

Conditional List-Pattern Probability**:** This feature was designed to eliminate the typical non-entailing cases of co-hyponyms (words sharing the same hypernym), which nevertheless tend to co-occur in entailment patterns. We therefore also check for pairs' occurrences in lists, using appropriate list patterns, expecting that correct entailment pairs would not co-occur in lists. The probability estimate, calculated like the previous one, is expected to be a negative feature for the learning model.

Relation Direction Ratio: The value of this feature is the ratio between the overall number of pattern matches for the pair and the number of pattern matches for the reversed pair (a pair created with the same terms in the opposite entailment direction). We found that this feature strongly correlates with entailment likelihood. Interestingly, it does not deteriorate performance for synonymous pairs.

Distributional Similarity Score: The *GD04* similarity score of the pair was used as a feature. We

PATTERN-BASED	DISTRIBU-TIONAL	TOTAL
1186	1420	2350

Table 2: The numbers of distinct entailment pair candidates obtained for the test words by each of the methods, and when combined.

also attempted adding Lin's (1998) similarity scores but they appeared to be redundant.

Intersection Feature: A binary feature indicating candidate pairs acquired by both methods, which was found to indicate higher entailment likelihood.

In summary, the above feature types utilize mutually complementary pattern-based and distributional information. Using cross validation over the training set we verified that each feature makes marginal contribution to performance when added on top of the remaining features.

3.4 Training and Classification

In order to systematically integrate different feature types we used the state-of-the-art supervised classifier SVM^{light} (Joachims, 1999) for entailment pair classification. Using 10-fold cross-validation over the training set we obtained the SVM configuration that yields an optimal micro-averaged F1 score. Through this optimization we chose the RBF kernel function and obtained optimal values for the *J*, *C* and the RBF's *Gamma* parameters. The candidate test pairs classified as correct entailments constitute the output of our integrated method.

4 Empirical Results

4.1 Data Set and Annotation

We utilized the experimental data set from Geffet and Dagan (2004). The dataset includes the similarity lists calculated by *GD04* for a sample of 30 target (common) nouns, computed from an 18 million word subset of the Reuters corpus[5]. We randomly picked a small set of 10 terms for training, leaving the remaining 20 terms for testing. Then, the set of entailment pair candidates for all nouns was created by applying the filtering method of Section 3.2.2 to the distributional similarity lists, and by extracting pattern-based candidates from the web as described in Section 3.2.1.

METHOD	P	R	F
Pattern-based	0.44	0.61	0.51
Distributional Similarity	0.33	0.53	0.40
Naïve Combination	0.36	1.00	0.53
Integrated	0.57	0.69	0.62

Table 3: Precision, Recall and F1 figures for the test words under each method.

Gold standard annotations for entailment pairs were created by three judges. The judges were guided to annotate as "Correct" the pairs conforming to the lexical entailment definition, which was reflected in two operational tests: i) *Word meaning entailment*: whether the meaning of the first (entailing) term implies the meaning of the second (entailed) term under some common sense of the two terms; and ii) *Substitutability*: whether the first term can substitute the second term in some natural contexts, such that the meaning of the modified context entails the meaning of the original one. The obtained Kappa values (varying between 0.7 and 0.8) correspond to *substantial agreement* on the task.

4.2 Results

The numbers of candidate entailment pairs collected for the test terms are shown in Table 2. These figures highlight the markedly complementary yield of the two acquisition approaches, where only about 10% of all candidates were identified by both methods. On average, 120 candidate entailment pairs were acquired for each target term.

The SVM classifier was trained on a quite small annotated sample of 700 candidate entailment pairs of the 10 training terms. Table 3 presents comparative results for the classifier, for each of the two sets of candidates produced by each method alone, and for the union of these two sets (referred as *Naïve Combination*). The results were computed for an annotated random sample of about 400 candidate entailment pairs of the test terms. Following common pooling evaluations in Information Retrieval, recall is calculated relatively to the total number of correct entailment pairs acquired by both methods together.

[5] Reuters Corpus, Volume 1, English Language, 1996-08-20 to 1997-08-19.

Pattern-based	Distributional
abduction → abuse	assault ↔ abuse
government → organization	government ↔ administration
drug therapy → treatment	budget deficit →gap
gap → hazard*	broker → analyst*
management → issue*	government → parliament*

Table 4: Typical entailment pairs acquired by the integrated method, illustrating Section 4.3. The columns specify the method that produced the candidate pair. Asterisk indicates a non-entailing pair.

The first two rows of the table show quite moderate precision and recall for the candidates of each separate method. The next row shows the great impact of method combination on recall, relative to the amount of correct entailment pairs found by each method alone, validating the complementary yield of the approaches. The integrated classifier, applied to the combined set of candidates, succeeds to increase precision substantially by 21 points (a relative increase of almost 60%), which is especially important for many precision-oriented applications like Information Retrieval and Question Answering. The precision increase comes with the expense of some recall, yet having F1 improved by 9 points. The integrated method yielded on average about 30 correct entailments per target term. Its classification *accuracy* (percent of correct classifications) reached 70%, which nearly doubles the naïve combination's accuracy.

It is impossible to directly compare our results with those of other works on lexical semantic relationships acquisition, since the particular task definition and dataset are different. As a rough reference point, our result figures do match those of related papers reviewed in Section 2, while we notice that our setting is relatively more difficult since we excluded the easier cases of proper nouns. (Geffet and Dagan, 2005), who exploited the distributional similarity approach over the web to address the same task as ours, obtained higher precision but substantially lower recall, considering only distributional candidates. Further research is suggested to investigate integrating their approach with ours.

4.3 Analysis and Discussion

Analysis of the data confirmed that the two methods tend to discover different types of relations. As expected, the distributional similarity method contributed most (75%) of the synonyms that were correctly classified as mutually entailing pairs (e.g. *assault* ↔ *abuse* in Table 4). On the other hand, about 80% of all correctly identified hyponymy relations were produced by the pattern-based method (e.g. *abduction* → *abuse*).

The integrated method provides a means to determine the entailment direction for distributional similarity candidates which by construction are non-directional. Thus, amongst the (non-synonymous) distributional similarity pairs classified as entailing, the direction of 73% was correctly identified. In addition, the integrated method successfully filters 65% of the non-entailing co-hyponym candidates (hyponyms of the same hypernym), most of them originated in the distributional candidates, which is a large portion (23%) of all correctly discarded pairs. Consequently, the precision of distributional similarity candidates approved by the integrated system was nearly doubled, indicating the additional information that patterns provide about distributionally similar pairs.

Yet, several error cases were detected and categorized. First, many non-entailing pairs are context-dependent, such as a *gap* which might constitute a *hazard* in some particular contexts, even though these words do not entail each other in their general meanings. Such cases are more typical for the pattern-based approach, which is sometimes permissive with respect to the relationship captured and may also extract candidates from a relatively small number of pattern occurrences. Second, synonyms tend to appear less frequently in patterns. Consequently, some synonymous pairs discovered by distributional similarity were rejected due to insufficient pattern matches. Anecdotally, some typos and spelling alternatives, like *privatization* ↔ *privatisation*, are also included in this category as they never co-occur in patterns.

In addition, a large portion of errors is caused by pattern ambiguity. For example, the pattern "NP_1, *a|an* NP_2", ranked among the top IS-A patterns by (Pantel et al., 2004), can represent both apposition (entailing) and a list of co-hyponyms (non-entailing). Finally, some misclassifications can be attributed to technical web-based processing errors and to corpus data sparseness.

5 Conclusion

The main contribution of this paper is a novel integration of the pattern-based and distributional approaches for lexical semantic acquisition, applied to lexical entailment. Our investigation highlights the complementary nature of the two approaches and the information they provide. Notably, it is possible to extract pattern-based information that complements the weaker evidence of distributional similarity. Supervised learning was found effective for integrating the different information types, yielding noticeably improved performance. Indeed, our analysis reveals that the integrated approach helps eliminating many error cases typical to each method alone. We suggest that this line of research may be investigated further to enrich and optimize the learning processes and to address additional lexical relationships.

Acknowledgement

We wish to thank Google for providing us with an extended quota for search queries, which made this research feasible.

References

Berland, Matthew and Charniak, Eugene. 1999. Finding parts in very large corpora. In Proc. of ACL-99. Maryland, USA.

Chklovski, Timothy and Patrick Pantel. 2004. VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In Proc. of EMNLP-04. Barcelona, Spain.

Dagan, Ido, Oren Glickman and Bernardo Magnini. 2005. The PASCAL Recognizing Textual Entailment Challenge. In Proc. of the PASCAL Challenges Workshop for Recognizing Textual Entailment. Southampton, U.K.

Etzioni, Oren, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D.S. Weld, and A. Yates. 2004. Web-scale information extraction in KnowItAll. In Proc. of WWW-04. NY, USA.

Geffet, Maayan and Ido Dagan. 2004. Feature Vector Quality and Distributional Similarity. In Proc. of COLING-04. Geneva, Switzerland.

Geffet, Maayan and Ido Dagan. 2005. The Distributional Inclusion Hypothesis and Lexical Entailment. In Proc of ACL-05. Michigan, USA.

Harris, Zelig S. 1968. Mathematical Structures of Language. Wiley.

Hearst, Marti. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In Proc. of COLING-92. Nantes, France.

Joachims, Thorsten. 1999. Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press.

Lin, Dekang. 1998. Automatic Retrieval and Clustering of Similar Words. In Proc. of COLING–ACL98, Montreal, Canada.

Lin, Dekang, Shaojun Zhao, Lijuan Qin, and Ming Zhou. 2003. Identifying synonyms among distributionally similar words. In Proc. of IJCAI-03. Acapulco, Mexico.

Pantel, Patrick, Deepak Ravichandran, and Eduard Hovy. 2004. Towards Terascale Semantic Acquisition. In Proc. of COLING-04. Geneva, Switzerland.

Pantel, Patrick and Deepak Ravichandran. 2004. Automatically Labeling Semantic Classes. In Proc. of HLT/NAACL-04. Boston, MA.

Ravichandran, Deepak and Eduard Hovy. 2002. Learning Surface Text Patterns for a Question Answering System. In Proc. of ACL-02. Philadelphia, PA.

Riloff, Ellen and Jessica Shepherd. 1997. A corpus-based approach for building semantic lexicons. In Proc. of EMNLP-97. RI, USA.

Riloff, Ellen and Rosie Jones. 1999. Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. In Proc. of AAAI-99. Florida, USA.

Shinzato, Kenji and Kentaro Torisawa. 2004. Acquiring Hyponymy Relations from Web Documents. In Proc. of HLT/NAACL-04. Boston, MA.

Sundblad, H. Automatic Acquisition of Hyponyms and Meronyms from Question Corpora. 2002. In Proc. of the ECAI-02 Workshop on Natural Language Processing and Machine Learning for Ontology Engineering. Lyon, France.

Weeds, Julie, David Weir, and Diana McCarthy. 2004. Characterizing Measures of Lexical Distributional Similarity. In Proc. of COLING-04. Geneva, Switzerland.

Yangarber, Roman, Ralph Grishman, Pasi Tapanainen and Silja Huttunen. 2000. Automatic Acquisition of Domain Knowledge for Information Extraction. In Proc. of COLING-00. Saarbrücken, Germany.