Fully Unsupervised Discovery of Concept-Specific Relationships by Web Mining

Dmitry Davidov
ICNC
The Hebrew University
Jerusalem 91904, Israel
dmitry@alice.nc.huji.ac.il

Ari Rappoport
Institute of Computer Science
The Hebrew University
Jerusalem 91904, Israel
www.cs.huji.ac.il/∼arir

Moshe Koppel
Dept. of Computer Science
Bar-Ilan University
Ramat-Gan 52900, Israel
koppel@cs.biu.ac.il

Abstract

We present a web mining method for discovering and enhancing relationships in which a specified concept (word class) participates. We discover a whole range of relationships focused on the given concept, rather than generic known relationships as in most previous work. Our method is based on clustering patterns that contain concept words and other words related to them. We evaluate the method on three different rich concepts and find that in each case the method generates a broad variety of relationships with good precision.

1 Introduction

The huge amount of information available on the web has led to a flurry of research on methods for automatic creation of structured information from large unstructured text corpora. The challenge is to create as much information as possible while providing as little input as possible.

A lot of this research is based on the initial insight (Hearst, 1992) that certain lexical patterns ('X is a country') can be exploited to automatically generate hyponyms of a specified word. Subsequent work (to be discussed in detail below) extended this initial idea along two dimensions.

One objective was to require as small a user-provided initial seed as possible. Thus, it was observed that given one or more such lexical patterns, a corpus could be used to generate examples of hyponyms that could then, in turn, be exploited to generate more lexical patterns. The larger and more reliable sets of patterns thus generated resulted in larger and more precise sets of hyponyms and vice versa. The initial step of the resulting alternating bootstrap process – the user-provided input – could just as well consist of examples of hyponyms as of lexical patterns.

A second objective was to extend the information that could be learned from the process beyond hyponyms of a given word. Thus, the approach was extended to finding lexical patterns that could produce synonyms and other standard lexical relations. These relations comprise all those words that stand in some known binary relation with a specified word.

In this paper, we introduce a novel extension of this problem: given a particular concept (initially represented by two seed words), discover relations in which it participates, without specifying their types in advance. We will generate a concept class and a variety of natural binary relations involving that class.

An advantage of our method is that it is particularly suitable for web mining, even given the restrictions on query amounts that exist in some of today's leading search engines.

The outline of the paper is as follows. In the next section we will define more precisely the problem we intend to solve. In section 3, we will consider related work. In section 4 we will provide an overview of our solution and in section 5 we will consider the details of the method. In section 6 we will illustrate and evaluate the results obtained by our method. Finally, in section 7 we will offer some conclusions and considerations for further work.

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 232–239,
Prague, Czech Republic, June 2007.

2 Problem Definition

In several studies (e.g., Widdows and Dorow, 2002; Pantel et al, 2004; Davidov and Rappoport, 2006) it has been shown that relatively unsupervised and language-independent methods could be used to generate many thousands of sets of words whose semantics is similar in some sense. Although examination of any such set invariably makes it clear why these words have been grouped together into a single concept, it is important to emphasize that the method itself provides no explicit concept definition; in some sense, the implied class is in the eye of the beholder. Nevertheless, both human judgment and comparison with standard lists indicate that the generated sets correspond to concepts with high precision.

We wish now to build on that result in the following way. Given a large corpus (such as the web) and two or more examples of some concept X, automatically generate examples of one or more relations $R \subset X \times Y$, where Y is some concept and R is some binary relationship between elements of X and elements of Y.

We can think of the relations we wish to generate as bipartite graphs. Unlike most earlier work, the bipartite graphs we wish to generate might be one-to-one (for example, countries and their capitals), many-to-one (for example, countries and the regions they are in) or many-to-many (for example, countries and the products they manufacture). For a given class X, we would like to generate not one but possibly many different such relations.

The only input we require, aside from a corpus, is a small set of examples of some class. However, since such sets can be generated in entirely unsupervised fashion, our challenge is effectively to generate relations directly from a corpus given no additional information of any kind. The key point is that we do not in any manner specify in advance what types of relations we wish to find.

3 Related Work

As far as we know, no previous work has directly addressed the discovery of generic binary relations in an unrestricted domain without (at least implicitly) pre-specifying relationship types. Most related work deals with discovery of hypernymy (Hearst, 1992; Pantel et al, 2004), synonymy (Roark and Charniak, 1998; Widdows and Dorow, 2002; Davidov and Rappoport, 2006) and meronymy (Berland and Charniak, 1999).

In addition to these basic types, several studies deal with the discovery and labeling of more specific relation sub-types, including inter-verb relations (Chklovski and Pantel, 2004) and noun-compound relationships (Moldovan et al, 2004).

Studying relationships between tagged named entities, (Hasegawa et al, 2004; Hassan et al, 2006) proposed unsupervised clustering methods that assign given (or semi-automatically extracted) sets of pairs into several clusters, where each cluster corresponds to one of a known relationship type. These studies, however, focused on the classification of pairs that were either given or extracted using some supervision, rather than on discovery and definition of which relationships are actually in the corpus.

Several papers report on methods for using the web to discover instances of binary relations. However, each of these assumes that the relations themselves are known in advance (implicitly or explicitly) so that the method can be provided with seed patterns (Agichtein and Gravano, 2000; Pantel et al, 2004), pattern-based rules (Etzioni et al, 2004), relation keywords (Sekine, 2006), or word pairs exemplifying relation instances (Pasca et al, 2006; Alfonseca et al, 2006; Rosenfeld and Feldman, 2006).

In some recent work (Strube and Ponzetto, 2006), it has been shown that related pairs can be generated without pre-specifying the nature of the relation sought. However, this work does not focus on differentiating among different relations, so that the generated relations might conflate a number of distinct ones.

It should be noted that some of these papers utilize language and domain-dependent preprocessing including syntactic parsing (Suchanek et al, 2006) and named entity tagging (Hasegawa et al, 2004), while others take advantage of handcrafted databases such as WordNet (Moldovan et al, 2004; Costello et al, 2006) and Wikipedia (Strube and Ponzetto, 2006).

Finally, (Turney, 2006) provided a pattern distance measure which allows a fully unsupervised measurement of relational similarity between two pairs of words; however, relationship types were not discovered explicitly.

4 Outline of the Method

We will use two concept words contained in a concept class C to generate a collection of distinct relations in which C participates. In this section we offer a brief overview of our method.

Step 1: Use a seed consisting of two (or more) example words to automatically obtain other examples that belong to the same class. Call these *concept words*. (For instance, if our example words were *France* and *Angola*, we would generate more country names.)

Step 2: For each concept word, collect instances of contexts in which the word appears together with one other content word. Call this other word a *target word* for that concept word. (For example, for *France* we might find 'Paris is the capital of France'. *Paris* would be a target word for *France*.)

Step 3: For each concept word, group the contexts in which it appears according to the target word that appears in the context. (Thus 'X is the capital of Y' would likely be grouped with 'Y's capital is X'.)

Step 4: Identify similar context groups that appear across many different concept words. Merge these into a single concept-word-independent cluster. (The group including the two contexts above would appear, with some variation, for other countries as well, and all these would be merged into a single cluster representing the relation *capital-of(X,Y)*.)

Step 5: For each cluster, output the relation consisting of all <concept word, target word> pairs that appear together in a context included in the cluster. (The cluster considered above would result in a set of pairs consisting of a country and its capital. Other clusters generated by the same seed might include countries and their languages, countries and the regions in which they are located, and so forth.)

5 Details of the Method

In this section we consider the details of each of the above-enumerated steps. It should be noted that each step can be performed using standard web searches; no special pre-processed corpus is required.

5.1 Generalizing the seed

The first step is to take the seed, which might consist of as few as two concept words, and generate many (ideally, all, when the concept is a closed set of words) members of the class to which they belong. We do this as follows, essentially implementing a simplified version of the method of Davidov and Rappoport (2006). For any pair of seed words S_i and S_j, search the corpus for word patterns of the form S_iHS_j, where H is a high-frequency word in the corpus (we used the 100 most frequent words in the corpus). Of these, we keep all those patterns, which we call *symmetric patterns*, for which S_jHS_i is also found in the corpus. Repeat this process to find symmetric patterns with any of the structures $HSHS$, $SHSH$ or $SHHS$. It was shown in (Davidov and Rappoport, 2006) that pairs of words that often appear together in such symmetric patterns tend to belong to the same class (that is, they share some notable aspect of their semantics). Other words in the class can thus be generated by searching a sub-corpus of documents including at least two concept words for those words X that appear in a sufficient number of instances of both the patterns S_iHX and XHS_i, where S_i is a word in the class. The same can be done for the other three pattern structures. The process can be bootstrapped as more words are added to the class.

Note that our method differs from that of Davidov and Rappoport (2006) in that here we provide an initial seed pair, representing our target concept, while there the goal is grouping of as many words as possible into concept classes. The focus of our paper is on relations involving a specific concept.

5.2 Collecting contexts

For each concept word S, we search the corpus for distinct contexts in which S appears. (For our purposes, a context is a window with exactly five words or punctuation marks before or after the concept word; we choose 10,000 of these, if available.) We call the aggregate text found in all these context windows the S-corpus.

From among these contexts, we choose all patterns of the form $H_1SH_2XH_3$ or $H_1XH_2SH_3$, where:

- X is a word that appears with frequency below f_1 in the S-corpus and that has sufficiently high pointwise mutual information with S. We use these two criteria to ensure that X is a content word and that it is related to S. The lower the threshold f_1, the less noise we allow in, though possibly at the expense of recall. We used $f_1 = 1,000$ occurrences per million words.

- H_2 is a string of words each of which occurs with frequency above f_2 in the S-corpus. We want H_2 to consist mainly of words common in the context of S in order to restrict patterns to those that are somewhat generic. Thus, in the context of countries we would like to retain words like *capital* while eliminating more specific words that are unlikely to express generic patterns. We used $f_2 = 100$ occurrences per million words (there is room here for automatic optimization, of course).

- H_1 and H_3 are either punctuation or words that occur with frequency above f_3 in the S-corpus. This is mainly to ensure that X and S aren't fragments of multi-word expressions. We used $f_3 = 100$ occurrences per million words.

- We call these patterns, *S-patterns* and we call X the *target* of the S-pattern. The idea is that S and X very likely stand in some fixed relation to each other where that relation is captured by the S-pattern.

5.3 Grouping S-patterns

If S is in fact related to X in some way, there might be a number of S-patterns that capture this relationship. For each X, we group all the S-patterns that have X as a target. (Note that two S-patterns with two different targets might be otherwise identical, so that essentially the same pattern might appear in two different groups.) We now merge groups with large (more than 2/3) overlap. We call the resulting groups, *S-groups.*

5.4 Identifying pattern clusters

If the S-patterns in a given S-group actually capture some relationship between S and the target, then one would expect that similar groups would appear for a multiplicity of concept words S. Suppose that we have S-groups for three different concept words S such that the pairwise overlap among the three groups is more than 2/3 (where for this purpose two patterns are deemed identical if they differ only at S and X). Then the set of patterns that appear in two or three of these S-groups is called a *cluster core.* We now group all patterns in other S-groups that have an overlap of more than 2/3 with the cluster core into a candidate pattern pool P. The set of all patterns in P that appear in at least two S-groups (among those that formed P) *pattern cluster.* A pattern cluster that has patterns instantiated by at least half of the concept words is said to represent a relation.

5.5 Refining relations

A relation consists of pairs (S, X) where S is a concept word and X is the target of some S-pattern in a given pattern cluster. Note that for a given S, there might be one or many values of X satisfying the relation. As a final refinement, for each given S, we rank all such X according to pointwise mutual information with S and retain only the highest 2/3. If most values of S have only a single corresponding X satisfying the relation and the rest have none, we try to automatically fill in the missing values by searching the corpus for relevant S-patterns for the missing values of S. (In our case the corpus is the web, so we perform additional clarifying queries.)

Finally, we delete all relations in which all concept words are related to most target words and all relations in which the concept words and the target words are identical. Such relations can certainly be of interest (see Section 7), but are not our focus in this paper.

5.6 Notes on required Web resources

In our implementation we use the Google search engine. Google restricts individual users to 1,000 queries per day and 1,000 pages per query. In each stage we conducted queries iteratively, each time downloading all 1,000 documents for the query.

In the first stage our goal was to discover symmetric relationships from the web and consequently discover additional concept words. For queries in this stage of our algorithm we invoked two requirements.

First, the query should contain at least two concept words. This proved very effective in reducing ambiguity. Thus of 1,000 documents for the query *bass*, 760 deal with music, while if we add to the query a second word from the intended concept (e.g., *barracuda*), then none of the 1,000 documents deal with music and the vast majority deal with fish, as intended.

Second, we avoid doing overlapping queries. To do this we used Google's ability to exclude from search results those pages containing a given term (in our case, one of the concept words).

We performed up to 300 different queries for individual concepts in the first stage of our algorithm.

In the second stage, we used web queries to assemble S-corpora. On average, about 1/3 of the concept words initially lacked sufficient data and we performed up to twenty additional queries for each rare concept word to fill its corpus.

In the last stage, when clusters are constructed, we used web queries for filling missing pairs of one-to-one or several-to-several relationships. The total number of filling queries for a specific concept was below 1,000, and we needed only the first results of these queries. Empirically, it took between 0.5 to 6 day limits (i.e., 500–6,000 queries) to extract relationships for a concept, depending on its size (the number of documents used for each query was at most 100). Obviously this strategy can be improved by focused crawling from primary Google hits, which can drastically reduce the required number of queries.

6 Evaluation

In this section we wish to consider the variety of relations that can be generated by our method from a given seed and to measure the quality of these relations in terms of their precision and recall.

With regard to precision, two claims are being made. One is that the generated relations correspond to identifiable relations. The other claim is that to the extent that a generated relation can be reasonably identified, the generated pairs do indeed belong to the identified relation. (There is a small degree of circularity in this characterization but this is probably the best we can hope for.)

As a practical matter, it is extremely difficult to measure precision and recall for relations that have not been pre-determined in any way. For each generated relation, authoritative resources must be marshaled as a gold standard. For purposes of evaluation, we ran our algorithm on three representative domains – countries, fish species and star constellations – and tracked down gold standard resources (encyclopedias, academic texts, informative websites, etc) for the bulk of the relations generated in each domain.

This choice of domains allowed us to explore different aspects of algorithmic behavior. Country and constellation domains are both well defined and closed domains. However they are substantially different.

Country names is a relatively large domain which has very low lexical ambiguity, and a large number of potentially useful relations. The main challenge in this domain was to capture it well.

Constellation names, in contrast, are a relatively small but highly ambiguous domain. They are used in proper names, mythology, names of entertainment facilities etc. Our evaluation examined how well the algorithm can deal with such ambiguity.

The fish domain contains a very high number of members. Unlike countries, it is a semi-open non-homogenous domain with a very large number of subclasses and groups. Also, unlike countries, it does not contain many proper nouns, which are empirically generally easier to identify in patterns. So the main challenge in this domain is to extract unblurred relationships and not to diverge from the domain during the concept acquisition phase.

We do not show here all-to-all relationships such as fish parts (common to all or almost all fish), because we focus on relationships that separate between members of the concept class, which are harder to acquire and evaluate.

6.1 Countries

Our seed consisted of two country names. The intended result for the first stage of the algorithm was a list of countries. There are 193 countries in the world (www.countrywatch.com) some of which have multiple names so that the total number of commonly used country names is 243. Of these, 223 names (comprising 180 countries) are character strings with no white space. Since we consider only single word names, these 223 are the names we hope to capture in this stage.

Using the seed words *France* and *Angola,* we obtained 202 country names (comprising 167 distinct countries) as well as 32 other names (consisting mostly of names of other geopolitical entities). Using the list of 223 single word countries as our gold standard, this gives precision of 0.90 and recall of 0.86. (Ten other seed pairs gave results ranging in precision: 0.86-0.93 and recall: 0.79-0.90.)

The second part of the algorithm generated a set of 31 binary relations. Of these, 25 were clearly identifiable relations many of which are shown in Table 1. Note that for three of these there are standard exhaustive lists against which we could measure both precision and recall; for the others shown, sources were available for measuring precision but no exhaustive list was available from which to measure recall, so we measured coverage (the number of countries for which at least one target concept is found as related).

Another eleven meaningful relations were generated for which we did not compute precision numbers. These include *celebrity-from, animal-of, lake-in, borders-on* and *enemy-of.* (The set of relations generated by other seed pairs differed only slightly from those shown here for *France* and *Angola.*)

6.2 Fish species

In our second experiment, our seed consisted of two fish species, *barracuda* and *bluefish.* There are 770 species listed in WordNet of which 447 names are character strings with no white space. The first stage of the algorithm returned 305 of the species listed in Wordnet, another 37 species not listed in Wordnet, as well as 48 other names (consisting mostly of other sea creatures). The second part of the algorithm generated a set of 15 binary relations all of which are meaningful. Those for which we could find some gold standard are listed in Table 2.

Other relations generated include *served-with, bait-for, food-type, spot-type,* and *gill-type.*

6.3 Constellations

Our seed consisted of two constellation names, *Orion* and *Cassiopeia.* There are 88 standard constellations (www.astro.wisc.edu) some of which have multiple names so that the total number of commonly used constellations is 98. Of these, 87 names (77 constellations) are strings with no white space.

Relationship *Sample pattern* (Sample pair)	Prec.	Rec/Cov
capital-of *in (x), capital of (y),* (Luanda, Angola)	0.92	R=0.79
language-spoken-in *to (x) or other (y) speaking* (Spain, Spanish)	0.92	R=0.60
in-region *throughout (x), from (y) to* (America, Canada)	0.73	R=0.71
city-in *west (x) – forecast for (y).* (England, London)	0.82	C=0.95
river-in *central (x), on the (y) river* (China, Haine)	0.92	C=0.68
mountain-range-in *the (x) mountains in (y) ,* (Chella, Angola)	0.77	C=0.69
sub-region-of *the (y) region of (x),* (Veneto, Italy)	0.81	C=0.81
industry-of *the (x) industry in (y) ,* (Oil, Russia)	0.70	C=0.90
island-in *, (x) island , (y) ,* (Bathurst, Canada)	0.98	C=0.55
president-of *president (x) of (y) has* (Bush, USA)	0.86	C=0.51
political-position-in *former (x) of (y) face* (President, Ecuador)	0.81	C=0.75
political-party-of *the (x) party of (y) ,* (Labour, England)	0.91	C=0.53
festival-of *the (x) festival, (y) ,* (Tanabata, Japan)	0.90	C=0.78
religious-denomination-of *the (x) church in (y) ,* (Christian, Rome)	0.80	C=0.62

Table 1: Results on seed { *France, Angola* }.

Relationship *Sample pattern* (Sample pair)	Prec.	Cov
region-found-in *best (x) fishing in (y) .* (Walleye, Canada)	0.83	0.80
sea-found-in *of (x) catches in the (y) sea* (Shark, Adriatic)	0.82	0.64
lake-found-in *lake (y) is famous for (x) ,* (Marion, Catfish)	0.79	0.51
habitat-of *, (x) and other (y) fish* (Menhaden, Saltwater)	0.78	0.92
also-called *. (y) , also called (x) ,* (Lemonfish, Ling)	0.91	0.58
eats *the (x) eats the (y) and* (Perch, Minnow)	0.90	0.85
color-of *the (x) was (y) color* (Shark, Gray)	0.95	0.85
used-for-food *catch (x) – best for (y) or* (Bluefish, Sashimi)	0.80	0.53
in-family *the (x) family , includes (y) ,* (Salmonid, Trout)	0.95	0.60

Table 2: Results on seed { *barracud, bluefish* }.

The first stage of the algorithm returned 81 constellation names (77 distinct constellations) as well as 38 other names (consisting mostly of names of individual stars). Using the list of 87 single word constellation names as our gold standard, this gives precision of 0.68 and recall of 0.93.

The second part of the algorithm generated a set of ten binary relations. Of these, one concerned travel and entertainment (constellations are quite popular as names of hotels and lounges) and another three were not interesting. Apparently, the requirement that half the constellations appear in a relation limited the number of viable relations since many constellations are quite obscure. The six interesting relations are shown in Table 3 along with precision and coverage.

7 Discussion

In this paper we have addressed a novel type of problem: given a specific concept, discover in fully unsupervised fashion, a range of relations in which it participates. This can be extremely useful for studying and researching a particular concept or field of study.

As others have shown as well, two concept words can be sufficient to generate almost the entire class to which the words belong when the class is well-defined. With the method presented in this paper, using no further user-provided information, we can, for a given concept, automatically generate a diverse collection of binary relations on this concept. These relations need not be pre-specified in any way. Results on the three domains we considered indicate that, taken as an aggregate, the relations that are generated for a given domain paint a rather clear picture of the range of information pertinent to that domain.

Moreover, all this was done using standard search engine methods on the web. No language-dependent tools were used (not even stemming); in fact, we reproduced many of our results using Google in Russian.

The method depends on a number of numerical parameters that control the subtle tradeoff between quantity and quality of generated relations. There is certainly much room for tuning of these parameters.

The concept and target words used in this paper are single words. Extending this to multiple-word expressions would substantially contribute to the applicability of our results.

In this research we effectively disregard many relationships of an all-to-all nature. However, such relationships can often be very useful for ontology construction, since in many cases they introduce strong connections between two different concepts. Thus, for fish we discovered that one of the all-to-all relationships captures a precise set of fish body parts, and another captures swimming verbs. Such relations introduce strong and distinct connections between the concept of fish and the concepts of fish-body-parts and swimming. Such connections may be extremely useful for ontology construction.

Relationship *Sample pattern* (Sample pair)	Prec.	Cov
nearby-constellation *constellation (x), near (y),* (Auriga, Taurus)	0.87	0.70
star-in *star (x) in (y) is* (Antares , Scorpius)	0.82	0.76
shape-of *, (x) is depicted as (y).* (Lacerta, Lizard)	0.90	0.55
abbreviated-as *. (x) abbr (y),* (Hidra, Hya)	0.93	0.90
cluster-types-in *famous (x) cluster in (y),* (Praesepe, Cancer)	0.92	1.00
location *, (x) is a (y) constellation* (Draco, Circumpolar)	0.82	0.70

Table 3: Results on seed { *Orion, Cassiopeia* }.

References

Agichtein, E., Gravano, L., 2000. Snowball: Extracting relations from large plain-text collections. Proceedings of the 5th ACM International Conference on Digital Libraries.

Alfonseca, E., Ruiz-Casado, M., Okumura, M., Castells, P., 2006. Towards large-scale non-taxonomic relation extraction: estimating the precision of rote extractors. Workshop on Ontology Learning and Population at COLING-ACL '06.

Berland, M., Charniak, E., 1999. Finding parts in very large corpora. ACL '99.

Chklovski T., Pantel P., 2004. VerbOcean: mining the web for fine-grained semantic verb relations. EMNLP '04.

Costello, F., Veale, T., Dunne, S., 2006. Using WordNet to automatically deduce relations between words in noun-noun compounds, COLING-ACL '06.

Davidov, D., Rappoport, A., 2006. Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words. COLING-ACL '06.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A., Shaked, T., Soderland, S., Weld, D., Yates, A., 2004. Methods for domain-independent information extraction from the web: an experimental comparison. AAAI '04.

Hasegawa, T., Sekine, S., Grishman, R., 2004. Discovering relations among named entities from large corpora. ACL '04.

Hassan, H., Hassan, A., Emam, O., 2006. unsupervised information extraction approach using graph mutual reinforcement. EMNLP '06.

Hearst, M., 1992. Automatic acquisition of hyponyms from large text corpora. COLING '92.

Moldovan, D., Badulescu, A., Tatu, M., Antohe, D., Girju, R., 2004. Models for the semantic classification of noun phrases. Workshop on Comput. Lexical Semantics at HLT-NAACL '04.

Pantel, P., Ravichandran, D., Hovy, E., 2004. Towards terascale knowledge acquisition. COLING '04.

Pasca, M., Lin, D., Bigham, J., Lifchits A., Jain, A., 2006. Names and similarities on the web: fact extraction in the fast lane. COLING-ACL '06.

Roark, B., Charniak, E., 1998. Noun-phrase co-occurrence statistics for semi-automatic semantic lexicon construction. ACL '98.

Rosenfeld B., Feldman, R.: URES : an unsupervised web relation extraction system. Proceedings, ACL '06 Poster Sessions.

Sekine, S., 2006 On-demand information extraction. COLING-ACL '06.

Strube, M., Ponzetto, S., 2006. WikiRelate! computing semantic relatedness using Wikipedia. AAAI '06.

Suchanek F. M., G. Ifrim, G. Weikum. 2006. LEILA: learning to extract information by linguistic analysis. Workshop on Ontology Learning and Population at COLING-ACL '06.

Turney, P., 2006. Expressing implicit semantic relations without supervision. COLING-ACL '06.

Widdows, D., Dorow, B., 2002. A graph model for unsupervised Lexical acquisition. COLING '02.