Unsupervised Discovery of Generic Relationships Using Pattern Clusters and its Evaluation by Automatically Generated SAT Analogy Questions

Dmitry Davidov
ICNC
Hebrew University of Jerusalem
dmitry@alice.nc.huji.ac.il

Ari Rappoport
Institute of Computer Science
Hebrew University of Jerusalem
arir@cs.huji.ac.il

Abstract

We present a novel framework for the discovery and representation of general semantic relationships that hold between lexical items. We propose that each such relationship can be identified with a cluster of patterns that captures this relationship. We give a fully unsupervised algorithm for pattern cluster discovery, which searches, clusters and merges high-frequency words-based patterns around randomly selected hook words. Pattern clusters can be used to extract instances of the corresponding relationships. To assess the quality of discovered relationships, we use the pattern clusters to automatically generate SAT analogy questions. We also compare to a set of known relationships, achieving very good results in both methods. The evaluation (done in both English and Russian) substantiates the premise that our pattern clusters indeed reflect relationships perceived by humans.

1 Introduction

Semantic resources can be very useful in many NLP tasks. Manual construction of such resources is labor intensive and susceptible to arbitrary human decisions. In addition, manually constructed semantic databases are not easily portable across text domains or languages. Hence, there is a need for developing semantic acquisition algorithms that are as unsupervised and language independent as possible.

A fundamental type of semantic resource is that of concepts (represented by sets of lexical items) and their inter-relationships. While there is relatively good agreement as to what concepts are

and which concepts should exist in a lexical resource, identifying types of important lexical relationships is a rather difficult task. Most established resources (e.g., WordNet) represent only the main and widely accepted relationships such as hypernymy and meronymy. However, there are many other useful relationships between concepts, such as noun-modifier and inter-verb relationships. Identifying and representing these explicitly can greatly assist various tasks and applications. There are already applications that utilize such knowledge (e.g., (Tatu and Moldovan, 2005) for textual entailment).

One of the leading methods in semantics acquisition is based on patterns (see e.g., (Hearst, 1992; Pantel and Pennacchiotti, 2006)). The standard process for pattern-based relation extraction is to start with hand-selected patterns or word pairs expressing a particular relationship, and iteratively scan the corpus for co-appearances of word pairs in patterns and for patterns that contain known word pairs. This methodology is semi-supervised, requiring pre-specification of the desired relationship or hand-coding initial seed words or patterns. The method is quite successful, and examining its results in detail shows that concept relationships are often being manifested by several different patterns.

In this paper, unlike the majority of studies that use patterns in order to find instances of given relationships, we use sets of patterns as the *definitions* of lexical relationships. We introduce *pattern clusters*, a novel framework in which each cluster corresponds to a relationship that can hold between the lexical items that fill its patterns' slots. We present a fully unsupervised algorithm to compute pat-

tern clusters, not requiring any, even implicit, pre-specification of relationship types or word/pattern seeds. Our algorithm does not utilize preprocessing such as POS tagging and parsing. Some patterns may be present in several clusters, thus indirectly addressing pattern ambiguity.

The algorithm is comprised of the following stages. First, we randomly select hook words and create a context corpus (hook corpus) for each hook word. Second, we define a meta-pattern using high frequency words and punctuation. Third, in each hook corpus, we use the meta-pattern to discover concrete patterns and target words co-appearing with the hook word. Fourth, we cluster the patterns in each corpus according to co-appearance of the target words. Finally, we merge clusters from different hook corpora to produce the final structure. We also propose a way to label each cluster by word pairs that represent it best.

Since we are dealing with relationships that are unspecified in advance, assessing the quality of the resulting pattern clusters is non-trivial. Our evaluation uses two methods: SAT tests, and comparison to known relationships. We used instances of the discovered relationships to automatically generate analogy SAT tests in two languages, English and Russian[1]. Human subjects answered these and real SAT tests. English grades were 80% for our test and 71% for the real test (83% and 79% for Russian), showing that our relationship definitions indeed reflect human notions of relationship similarity. In addition, we show that among our pattern clusters there are clusters that cover major known noun-compound and verb-verb relationships.

In the present paper we focus on the pattern cluster resource itself and how to evaluate its intrinsic quality. In (Davidov and Rappoport, 2008) we show how to *use* the resource for a known task of a totally different nature, classification of relationships between nominals (based on annotated data), obtaining superior results over previous work.

Section 2 discusses related work, and Section 3 presents the pattern clustering and labeling algorithm. Section 4 describes the corpora we used and the algorithm's parameters in detail. Sections 5 and

---

[1]Turney and Littman (2005) automatically answers SAT tests, while our focus is on generating them.

6 present SAT and comparison evaluation results.

2 Related Work

Extraction of relation information from text is a large sub-field in NLP. Major differences between pattern approaches include the relationship types sought (including domain restrictions), the degrees of supervision and required preprocessing, and evaluation method.

2.1 Relationship Types

There is a large body of related work that deals with discovery of basic relationship types represented in useful resources such as WordNet, including hypernymy (Hearst, 1992; Pantel et al., 2004; Snow et al., 2006), synonymy (Davidov and Rappoport, 2006; Widdows and Dorow, 2002) and meronymy (Berland and Charniak, 1999; Girju et al., 2006).

Since named entities are very important in NLP, many studies define and discover relations between named entities (Hasegawa et al., 2004; Hassan et al., 2006). Work was also done on relations between verbs (Chklovski and Pantel, 2004). There is growing research on relations between nominals (Moldovan et al., 2004; Girju et al., 2007).

2.2 Degree of Supervision and Preprocessing

While numerous studies attempt to discover one or more pre-specified relationship types, very little previous work has directly attempted the discovery of which main types of generic relationships actually exist in an unrestricted domain. Turney (2006) provided a pattern distance measure that allows a fully unsupervised measurement of relational similarity between two pairs of words; such a measure could in principle be used by a clustering algorithm in order to deduce relationship types, but this was not discussed. Unlike (Turney, 2006), we do not perform any pattern ranking. Instead we produce (possibly overlapping) hard clusters, where each pattern cluster represents a relationship discovered in the domain. Banko et al. (2007) and Rosenfeld and Feldman (2007) find relationship instances where the relationships are not specified in advance. They aim to find relationship instances rather than identify generic semantic relationships. Thus, their representation is very different from ours. In addition, (Banko et al., 2007) utilize supervised tools such

as a POS tagger and a shallow parser. Davidov et al. (2007) proposed a method for unsupervised discovery of concept-specific relations. That work, like ours, relies on pattern clusters. However, it requires initial word seeds and targets the discovery of relationships specific for some given concept, while we attempt to discover and define generic relationships that exist in the entire domain.

Studying relationships between tagged named entities, (Hasegawa et al., 2004; Hassan et al., 2006) proposed unsupervised clustering methods that assign given sets of pairs into several clusters, where each cluster corresponds to one of a known set of relationship types. Their classification setting is thus very different from our unsupervised discovery one.

Several recent papers discovered relations on the web using seed patterns (Pantel et al., 2004), rules (Etzioni et al., 2004), and word pairs (Pasca et al., 2006; Alfonseca et al., 2006). The latter used the notion of hook which we also use in this paper. Several studies utilize some preprocessing, including parsing (Hasegawa et al., 2004; Hassan et al., 2006) and usage of syntactic (Suchanek et al., 2006) and morphological (Pantel et al., 2004) information in patterns. Several algorithms use manually-prepared resources, including WordNet (Moldovan et al., 2004; Costello et al., 2006) and Wikipedia (Strube and Ponzetto, 2006). In this paper, we do not utilize any language-specific preprocessing or any other resources, which makes our algorithm relatively easily portable between languages, as we demonstrate in our bilingual evaluation.

2.3 Evaluation Method

Evaluation for hypernymy and synonymy usually uses WordNet (Lin and Pantel, 2002; Widdows and Dorow, 2002; Davidov and Rappoport, 2006). For more specific lexical relationships like relationships between verbs (Chklovski and Pantel, 2004), nominals (Girju et al., 2004; Girju et al., 2007) or meronymy subtypes (Berland and Charniak, 1999) there is still little agreement which important relationships should be defined. Thus, there are more than a dozen different type hierarchies and tasks proposed for noun compounds (and nominals in general), including (Nastase and Szpakowicz, 2003; Girju et al., 2005; Girju et al., 2007).

There are thus two possible ways for a fair eval-uation. A study can develop its own relationship definitions and dataset, like (Nastase and Szpakowicz, 2003), thus introducing a possible bias; or it can accept the definition and dataset prepared by another work, like (Turney, 2006). However, this makes it impossible to work on new relationship types. Hence, when exploring very specific relationship types or very generic, but not widely accepted, types (like verb strength), many researchers resort to manual human-based evaluation (Chklovski and Pantel, 2004). In our case, where relationship types are not specified in advance, creating an unbiased benchmark is very problematic, so we rely on human subjects for relationship evaluation.

3 Pattern Clustering Algorithm

Our algorithm first discovers and clusters patterns in which a single ('hook') word participates, and then merges the resulting clusters to form the final structure. In this section we detail the algorithm. The algorithm utilizes several parameters, whose selection is detailed in Section 4. We refer to a pattern contained in our clusters (a pattern type) as a 'pattern' and to an occurrence of a pattern in the corpus (a pattern token) as a 'pattern instance'.

3.1 Hook Words and Hook Corpora

As a first step, we randomly select a set of hook words. Hook words were used in e.g. (Alfonseca et al., 2006) for extracting general relations starting from given seed word pairs. Unlike most previous work, our hook words are not provided in advance but selected randomly; the goal in those papers is to discover relationships between given word pairs, while we use hook words in order to discover relationships that generally occur in the corpus.

Only patterns in which a hook word actually participates will eventually be discovered. Hence, in principle we should select as many hook words as possible. However, words whose frequency is very high are usually ambiguous and are likely to produce patterns that are too noisy, so we do not select words with frequency higher than a parameter F_C. In addition, we do not select words whose frequency is below a threshold F_B, to avoid selection of typos and other noise that frequently appear on the web.

We also limit the total number N of hook words.

Our algorithm merges clusters originating from different hook words. Using too many hook words increases the chance that some of them belong to a noisy part in the corpus and thus lowers the quality of our resulting clusters.

For each hook word, we now create a hook corpus, the set of the contexts in which the word appears. Each context is a window containing W words or punctuation characters before and after the hook word. We avoid extracting text from clearly unformatted sentences and our contexts do not cross paragraph boundaries.

The size of each hook corpus is much smaller than that of the whole corpus, easily fitting into main memory; the corpus of a hook word occurring h times in the corpus contains at most $2hW$ words. Since most operations are done on each hook corpus separately, computation is very efficient.

Note that such context corpora can in principle be extracted by focused querying on the web, making the system dynamically scalable. It is also possible to restrict selection of hook words to a specific domain or word type, if we want to discover only a desired subset of existing relationships. Thus we could sample hook words from nouns, verbs, proper names, or names of chemical compounds if we are only interested in discovering relationships between these. Selecting hook words randomly allows us to avoid using any language-specific data at this step.

3.2 Pattern Specification

In order to reduce noise and to make the computation more efficient, we did not consider all contexts of a hook word as pattern candidates, only contexts that are instances of a specified meta-pattern type. Following (Davidov and Rappoport, 2006), we classified words into high-frequency words (HFWs) and content words (CWs). A word whose frequency is more (less) than F_H (F_C) is considered to be a HFW (CW). Unlike (Davidov and Rappoport, 2006), we consider all punctuation characters as HFWs. Our patterns have the general form

$$\text{[Prefix] } CW_1 \text{ [Infix] } CW_2 \text{ [Postfix]}$$

where Prefix, Infix and Postfix contain only HFWs. To reduce the chance of catching CW_i's that are parts of a multiword expression, we require Prefix and Postfix to have at least one word (HFW), while Infix is allowed to contain any number of HFWs (but recall that the total length of a pattern is limited by window size). A pattern example is *'such X as Y and'*. During this stage we only allow single words to be in CW slots[2].

3.3 Discovery of Target Words

For each of the hook corpora, we now extract all pattern instances where one CW slot contains the hook word and the other CW slot contains some other ('target') word. To avoid the selection of common words as target words, and to avoid targets appearing in pattern instances that are relatively fixed multiword expressions, we sort all target words in a given hook corpus by pointwise mutual information between hook and target, and drop patterns obtained from pattern instances containing the lowest and highest L percent of target words.

3.4 Local Pattern Clustering

We now have for each hook corpus a set of patterns. All of the corresponding pattern instances share the hook word, and some of them also share a target word. We cluster patterns in a two-stage process. First, we group in clusters all patterns whose instances share the same target word, and ignore the rest. For each target word we have a single pattern cluster. Second, we merge clusters that share more than S percent of their patterns. A pattern can appear in more than a single cluster. Note that clusters contain pattern *types*, obtained through examining pattern *instances*.

3.5 Global Cluster Merging

The purpose of this stage is to create clusters of patterns that express generic relationships rather than ones specific to a single hook word. In addition, the technique used in this stage reduces noise. For each created cluster we will define *core* patterns and *unconfirmed* patterns, which are weighed differently during cluster labeling (see Section 3.6). We merge clusters from different hook corpora using the following algorithm:

1. Remove all patterns originating from a single hook corpus.

---

[2]While for pattern clusters creation we use only single words as CWs, later during evaluation we allow multiword expressions in CW slots of previously acquired patterns.

2. Mark all patterns of all present clusters as unconfirmed.
3. While there exists some cluster C_1 from corpus D_X containing only unconfirmed patterns:
   (a) Select a cluster with a minimal number of patterns.
   (b) For each corpus D different from D_X:
      i. Scan D for clusters C_2 that share at least S percent of their patterns, and all of their core patterns, with C_1.
      ii. Add all patterns of C_2 to C_1, setting all shared patterns as core and all others as unconfirmed.
      iii. Remove cluster C_2.
   (c) If all of C_1's patterns remain unconfirmed remove C_1.
4. If several clusters have the same set of core patterns merge them according to rules (i,ii).

We start from the smallest clusters because we expect these to be more precise; the best patterns for semantic acquisition are those that belong to small clusters, and appear in many different clusters. At the end of this algorithm, we have a set of pattern clusters where for each cluster there are two subsets, core patterns and unconfirmed patterns.

3.6 Labeling of Pattern Clusters

To label pattern clusters we define a HITS measure that reflects the affinity of a given word pair to a given cluster. For a given word pair (w_1, w_2) and cluster C with n core patterns P_{core} and m unconfirmed patterns P_{unconf},

$$Hits(C, (w_1, w_2)) =$$
$$|\{p; (w_1, w_2) \text{ appears in } p \in P_{core}\}| / n +$$
$$\alpha \times |\{p; (w_1, w_2) \text{ appears in } p \in P_{unconf}\}| / m.$$

In this formula, 'appears in' means that the word pair appears in instances of this pattern extracted from the original corpus or retrieved from the web during evaluation (see Section 5.2). Thus if some pair appears in most of patterns of some cluster it receives a high HITS value for this cluster. The top 5 pairs for each cluster are selected as its labels. $\alpha \in (0..1)$ is a parameter that lets us modify the relative weight of core and unconfirmed patterns.

4 Corpora and Parameters

In this section we describe our experimental setup, and discuss in detail the effect of each of the algorithms' parameters.

4.1 Languages and Corpora

The evaluation was done using corpora in English and Russian. The English corpus (Gabrilovich and Markovitch, 2005) was obtained through crawling the URLs in the Open Directory Project (dmoz.org). It contains about 8.2G words and its size is about 68GB of untagged plain text. The Russian corpus was collected over the web, comprising a variety of domains, including news, web pages, forums, novels and scientific papers. It contains 7.5G words of size 55GB untagged plain text. Aside from removing noise and sentence duplicates, we did not apply any text preprocessing or tagging.

4.2 Parameters

Our algorithm uses the following parameters: F_C, F_H, F_B, W, N, L, S and α. We used part of the Russian corpus as a development set for determining the parameters. On our development set we have tested various parameter settings. A detailed analysis of the involved parameters is beyond the scope of this paper; below we briefly discuss the observed qualitative effects of parameter selection. Naturally, the parameters are not mutually independent.

F_C (upper bound for content word frequency in patterns) influences which words are considered as hook and target words. More ambiguous words generally have higher frequency. Since content words determine the joining of patterns into clusters, the more ambiguous a word is, the noisier the resulting clusters. Thus, higher values of F_C allow more ambiguous words, increasing cluster recall but also increasing cluster noise, while lower ones increase cluster precision at the expense of recall.

F_H (lower bound for HFW frequency in patterns) influences the specificity of patterns. Higher values restrict our patterns to be based upon the few most common HFWs (like 'the', 'of', 'and') and thus yield patterns that are very generic. Lowering the values, we obtain increasing amounts of pattern clusters for more specific relationships. The value we use for F_H is lower than that used for F_C, in order to allow as HFWs function words of relatively low frequency (e.g., 'through'), while allowing as content words some frequent words that participate in meaningful relationships (e.g., 'game'). However, this way we may also introduce more noise.

F_B (lower bound for hook words) filters hook words that do not appear enough times in the corpus. We have found that this parameter is essential for removing typos and other words that do not qualify as hook words.

N (number of hook words) influences relationship coverage. With higher N values we discover more relationships roughly of the same specificity level, but computation becomes less efficient and more noise is introduced.

W (window size) determines the length of the discovered patterns. Lower values are more efficient computationally, but values that are too low result in drastic decrease in coverage. Higher values would be more useful when we allow our algorithm to support multiword expressions as hooks and targets.

L (target word mutual information filter) helps in avoiding using as targets common words that are unrelated to hooks, while still catching as targets frequent words that are related. Low L values decrease pattern precision, allowing patterns like *'give X please Y more'*, where X is the hook (e.g., 'Alex') and Y the target (e.g., 'some'). High values increase pattern precision at the expense of recall.

S (minimal overlap for cluster merging) is a clusters merge filter. Higher values cause more strict merging, producing smaller but more precise clusters, while lower values start introducing noise. In extreme cases, low values can start a chain reaction of total merging.

α (core vs. unconfirmed weight for HITS labeling) allows lower quality patterns to complement higher quality ones during labeling. Higher values increase label noise, while lower ones effectively ignore unconfirmed patterns during labeling.

In our experiments we have used the following values (again, determined using a development set) for these parameters: F_C: $1,000$ words per million (wpm); F_H: 100 wpm; F_B: 1.2 wpm; N: 500 words; W: 5 words; L: 30%; S: 2/3; α: 0.1.

5 SAT-based Evaluation

As discussed in Section 2, the evaluation of semantic relationship structures is non-trivial. The goal of our evaluation was to assess whether pattern clusters indeed represent meaningful, precise and different relationships. There are two complementary perspec-

tives that a pattern clusters quality assessment needs to address. The first is the quality (precision/recall) of individual pattern clusters: does each pattern cluster capture lexical item pairs of the same semantic relationship? does it recognize many pairs of the same semantic relationship? The second is the quality of the cluster set as whole: does the pattern clusters set allow identification of important known semantic relationships? do several pattern clusters describe the same relationship?

Manually examining the resulting pattern clusters, we saw that the majority of sampled clusters indeed clearly express an interesting specific relationship. Examples include familiar hypernymy clusters such as[3] { *'such X as Y'*, *'X such as Y'*, *'Y and other X'*,} with label (*pets, dogs*), and much more specific clusters like { *'buy Y accessory for X!'*, *'shipping Y for X'*, *'Y is available for X'*, *'Y are available for X'*, *'Y are available for X systems'*, *'Y for X'* }, labeled by (*phone, charger*). Some clusters contain overlapping patterns, like *'Y for X'*, but represent different relationships when examined as a whole.

We addressed the evaluation questions above using a SAT-like analogy test automatically generated from word pairs captured by our clusters (see below in this section). In addition, we tested coverage and overlap of pattern clusters with a set of 35 known relationships, and we compared our patterns to those found useful by other algorithms (the next section).

Quantitatively, the final number of clusters is 508 (470) for English (Russian), and the average cluster size is 5.5 (6.1) pattern types. 55% of the clusters had no overlap with other clusters.

5.1 SAT Analogy Choice Test

Our main evaluation method, which is also a useful application by itself, uses our pattern clusters to automatically generate SAT analogy questions. The questions were answered by human subjects.

We randomly selected 15 clusters. This allowed us to assess the precision of the whole cluster set as well as of the internal coherence of separate clusters (see below). For each cluster, we constructed a SAT analogy question in the following manner. The header of the question is a word pair that is one of the label pairs of the cluster. The five multiple

---

[3]For readability, we omit punctuations in Prefix and Postfix.

choice items include: (1) another label of the cluster (the 'correct' answer); (2) three labels of other clusters among the 15; and (3) a pair constructed by randomly selecting words from those making up the various cluster labels.

In our sample there were no word pairs assigned as labels to more than one cluster[4]. As a baseline for comparison, we have mixed these questions with 15 real SAT questions taken from English and Russian SAT analogy tests. In addition, we have also asked our subjects to write down one example pair of the same relationship for each question in the test.

As an example, from one of the 15 clusters we have randomly selected the label (*glass, water*). The correct answer selected from the same cluster was (*schoolbag, book*). The three pairs randomly selected from the other 14 clusters were (*war, death*), (*request, license*) and (*mouse, cat*). The pair randomly selected from a cluster not among the 15 clusters was (*milk, drink*). Among the subjects' proposals for this question were (*closet, clothes*) and (*wallet, money*).

We computed accuracy of SAT answers, and the correlation between answers for our questions and the real ones (Table 1). Three things are demonstrated about our system when humans are capable of selecting the correct answer. First, our clusters are internally coherent in the sense of expressing a certain relationship, because people identified that the pairs in the question header and in the correct answer exhibit the same relationship. Second, our clusters distinguish between different relationships, because the three pairs not expressing the same relationship as the header were not selected by the evaluators. Third, our cluster labeling algorithm produces results that are usable by people.

The test was performed in both English and Russian, with 10 (6) subjects for English (Russian). The subjects (biology and CS students) were not involved with the research, did not see the clusters, and did not receive any special training as preparation. Inter-subject agreement and Kappa were 0.82, 0.72 (0.9, 0.78) for English (Russian). As reported in (Turney, 2005), an average high-school SAT grade is 57. Table 1 shows the final English and Russian grade average for ours and real SAT questions.

---

[4]But note that a pair can certainly obtain a positive HITS value for several clusters.

	Our method	Real SAT	Correlation
English	**80%**	71%	0.85
Russian	**83%**	79%	0.88

Table 1: Pattern cluster evaluation using automatically generated SAT analogy choice questions.

We can see that for both languages, around 80% of the choices were correct (the random choice baseline is 20%). Our subjects are university students, so results higher than 57 are expected, as we can see from real SAT performance. The difference in grades between the two languages might be attributed to the presence of relatively hard and uncommon words. It also may result from the Russian test being easier because there is less verb-noun ambiguity in Russian.

We have observed a high correlation between true grades and ours, suggesting that our automatically generated test reflects the ability to recognize analogies and can be potentially used for automated generation of SAT-like tests.

The results show that our pattern clusters indeed mirror a human notion of relationship similarity and represent meaningful relationships. They also show that as intended, different clusters describe different relationships.

5.2 Analogy Invention Test

To assess recall of separate pattern clusters, we have asked subjects to provide (if possible) an additional pair for each SAT question. On each such pair we have automatically extracted a set of pattern instances that capture this pair by using automated web queries. Then we calculated the HITS value for each of the selected pairs and assigned them to clusters with highest HITS value. The numbers of pairs provided were 81 for English and 43 for Russian.

We have estimated precision for this task as macro-average of percentage of correctly assigned pairs, obtaining 87% for English and 82% for Russian (the random baseline of this 15-class classification task is 6.7%). It should be noted however that the human-provided additional relationship examples in this test are not random so it may introduce bias. Nevertheless, these results confirm that our pattern clusters are able to recognize new in-

30 Noun Compound Relationships		
	Avg. num of clusters	Overlap
Russian	1.8	0.046
English	1.7	0.059
5 Verb Verb Relationships		
Russian	1.4	0.01
English	1.2	0

Table 2: Patterns clusters discovery of known relationships.

stances of relationships of the same type.

6 Evaluation Using Known Information

We also evaluated our pattern clusters using relevant information reported in related work.

6.1 Discovery of Known Relationships

To estimate recall of our pattern cluster set, we attempted to estimate whether (at least) a subset of known relationships have corresponding pattern clusters. As a testing subset, we have used 35 relationships for both English and Russian. 30 relations are noun compound relationships as proposed in the (Nastase and Szpakowicz, 2003) classification scheme, and 5 relations are verb-verb relations proposed by (Chklovski and Pantel, 2004). We have manually created sets of 5 unambiguous sample pairs for each of these 35 relationships. For each such pair we have assigned the pattern cluster with best HITS value.

The middle column of Table 2 shows the average number of clusters per relationship. Ideally, if for each relationship all 5 pairs are assigned to the same cluster, the average would be 1. In the worst case, when each pair is assigned to a different cluster, the average would be 5. We can see that most of the pairs indeed fall into one or two clusters, successfully recognizing that similarly related pairs belong to the same cluster. The column on the right shows the overlap between different clusters, measured as the average number of shared pairs in two randomly selected clusters. The baseline in this case is essentially 5, since there are more than 400 clusters for 5 word pairs. We see a very low overlap between assigned clusters, which shows that these clusters indeed separate well between defined relations.

6.2 Discovery of Known Pattern Sets

We compared our clusters to lists of patterns reported as useful by previous papers. These lists included patterns expressing hypernymy (Hearst, 1992; Pantel et al., 2004), meronymy (Berland and Charniak, 1999; Girju et al., 2006), synonymy (Widdows and Dorow, 2002; Davidov and Rappoport, 2006), and verb strength + verb happensbefore (Chklovski and Pantel, 2004). In all cases, we discovered clusters containing all of the reported patterns (including their refinements with domain-specific prefix or postfix) and not containing patterns of competing relationships.

7 Conclusion

We have proposed a novel way to define and identify generic lexical relationships as clusters of patterns. Each such cluster is set of patterns that can be used to identify, classify or capture new instances of some unspecified semantic relationship. We showed how such pattern clusters can be obtained automatically from text corpora without any seeds and without relying on manually created databases or language-specific text preprocessing. In an evaluation based on an automatically created analogy SAT test we showed on two languages that pairs produced by our clusters indeed strongly reflect human notions of relation similarity. We also showed that the obtained pattern clusters can be used to recognize new examples of the same relationships. In an additional test where we assign labeled pairs to pattern clusters, we showed that they provide good coverage for known noun-noun and verb-verb relationships for both tested languages.

While our algorithm shows good performance, there is still room for improvement. It utilizes a set of constants that affect precision, recall and the granularity of the extracted cluster set. It would be beneficial to obtain such parameters automatically and to create a multilevel relationship hierarchy instead of a flat one, thus combining different granularity levels. In this study we applied our algorithm to a generic domain, while the same method can be used for more restricted domains, potentially discovering useful domain-specific relationships.

References

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

Banko, M., Cafarella, M. J. , Soderland, S., Broadhead, M., and Etzioni, O., 2007. Open information extraction from the Web. *IJCAI '07*.

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 Word-Net 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*.

Davidov, D., Rappoport, A. and Koppel, M., 2007. Fully unsupervised discovery of concept-specific relationships by Web mining. *ACL '07*.

Davidov, D., Rappoport, A., 2008. Classification of relationships between nominals using pattern clusters. *ACL '08*.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A., Shaked, T., Soderland, S., Weld, D., and Yates, A., 2004. Methods for domain-independent information extraction from the web: An experimental comparison. *AAAI 04*

Gabrilovich, E., Markovitch, S., 2005. Feature generation for text categorization using world knowledge. *IJCAI 2005*.

Girju, R., Giuglea, A., Olteanu, M., Fortu, O., Bolohan, O., and Moldovan, D., 2004. Support vector machines applied to the classification of semantic relations in nominalized noun phrases. *HLT/NAACL Workshop on Computational Lexical Semantics*.

Girju, R., Moldovan, D., Tatu, M., and Antohe, D., 2005. On the semantics of noun compounds. *Computer Speech and Language*, 19(4):479-496.

Girju, R., Badulescu, A., and Moldovan, D., 2006. Automatic discovery of part-whole relations. *Computational Linguistics*, 32(1).

Girju, R., Hearst, M., Nakov, P., Nastase, V., Szpakowicz, S., Turney, P., and Yuret, D., 2007. Task 04: Classification of semantic relations between nominal at SemEval 2007. *ACL '07 SemEval Workshop*.

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

Hassan, H., Hassan, A. and 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*

Lin, D., Pantel, P., 2002. Concept discovery from text. *COLING 02*.

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

Nastase, V., Szpakowicz, S., 2003. Exploring noun modifier semantic relations. *IWCS-5*.

Pantel, P., Pennacchiotti, M., 2006. Espresso: leveraging generic patterns for automatically harvesting semantic relations. *COLING-ACL 2006*.

Pantel, P., Ravichandran, D. and Hovy, E.H., 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*.

Rosenfeld, B., Feldman, R., 2007. Clustering for unsupervised relation identification. *CIKM '07*.

Snow, R., Jurafsky, D., Ng, A.Y., 2006. Semantic taxonomy induction from heterogeneous evidence. *COLING-ACL '06*.

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

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

Tatu, M., Moldovan, D., 2005. A semantic approach to recognizing textual entailment. *HLT/EMNLP 2005*.

Turney, P., 2005. Measuring semantic similarity by latent relational analysis. *IJCAI '05*.

Turney, P., Littman, M., 2005. Corpus-based learning of analogies and semantic selations. *Machine Learning*(60):1–3:251–278.

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*.