Superior and Efficient Fully Unsupervised Pattern-based Concept Acquisition Using an Unsupervised Parser

Dmitry Davidov[1] **Roi Reichart**[1] **Ari Rappoport**[2]

[1]ICNC , [2]Institute of Computer Science
Hebrew University of Jerusalem
{dmitry@alice.nc|roiri@cs|arir@cs}.huji.ac.il

Abstract

Sets of lexical items sharing a significant aspect of their meaning (*concepts*) are fundamental for linguistics and NLP. Unsupervised concept acquisition algorithms have been shown to produce good results, and are preferable over manual preparation of concept resources, which is labor intensive, error prone and somewhat arbitrary. Some existing concept mining methods utilize supervised language-specific modules such as POS taggers and computationally intensive parsers.

In this paper we present an efficient fully unsupervised concept acquisition algorithm that uses syntactic information obtained from a fully unsupervised parser. Our algorithm incorporates the bracketings induced by the parser into the meta-patterns used by a symmetric patterns and graph-based concept discovery algorithm. We evaluate our algorithm on very large corpora in English and Russian, using both human judgments and WordNet-based evaluation. Using similar settings as the leading fully unsupervised previous work, we show a significant improvement in concept quality and in the extraction of multiword expressions. Our method is the first to use fully unsupervised parsing for unsupervised concept discovery, and requires no language-specific tools or pattern/word seeds.

1 Introduction

Comprehensive lexical resources for many domains and languages are essential for most NLP applications. One of the most utilized types of such resources is a repository of *concepts*: sets of lexical items sharing a significant aspect of their meanings (e.g., types of food, tool names, etc).

While handcrafted concept databases (e.g., WordNet) are extensively used in NLP, manual compilation of such databases is labor intensive, error prone, and somewhat arbitrary. Hence, for many languages and domains great efforts have been made for automated construction of such databases from available corpora. While language-specific and domain-specific studies show significant success in development of concept discovery frameworks, the majority of domains and languages remain untreated. Hence there is a need for a framework that performs well for many diverse settings and is as unsupervised and language-independent as possible.

Numerous methods have been proposed for seed-based concept extraction where a set of concept patterns (or rules), or a small set of seed words for each concept, is provided as input to the concept acquisition system. However, even simple definitions for concepts are not always available.

To avoid requiring this type of input, a number of distributional and pattern-based methods have been proposed for fully unsupervised seed-less acquisition of concepts from text. Pattern-based algorithms were shown to obtain high quality results while being highly efficient in comparison to distributional methods. Such fully unsupervised methods do not incorporate any language-specific parsers or taggers, so can be successfully applied to diverse languages.

However, unsupervised pattern-based methods suffer from several weaknesses. Thus they are frequently restricted to single-word terms and are unable to discover multiword expressions in efficient and precise manner. They also usually ignore potentially useful part-of-speech and other syntactic information. In order to address these weaknesses, several studies utilize language-specific parsing or tagging systems in concept acquisition. Unfortunately, while improving results, this heavily affects the language- and domain- independence of such frameworks, and severely impacts efficiency since even shallow parsing is computationally demanding.

In this paper we present a method to utilize the information induced by unsupervised parsers in an unsupervised pattern-based concept discovery framework. With the recent development of fast fully unsupervised parsers, it is now possible to add parser-based information to lexical patterns while keeping the language-independence of the whole framework and still avoiding heavy computational costs. Specifically, we incorporate the bracketings induced by the parser into the meta-patterns used by a symmetric patterns and graph-based unsupervised concept discovery algorithm.

We performed a thorough evaluation on two English corpora (the BNC and a 68GB web corpus) and on a 33GB Russian corpus. Evaluations were done using both human judgments and WordNet, in similar settings as that of the leading unsupervised previous work. Our results show that utilization of unsupervised parser both improves the assignment of single-word terms to concepts and allows high-precision discovery and assignment of of multiword expressions to concepts.

2 Previous Work

Much work has been done on lexical acquisition of all sorts and the acquisition of concepts in particular. Concept acquisition methods differ in the type of corpus annotation and other human input used, and in their basic algorithmic approach. Some methods directly aim at concept acquisition, while the direct goal in some is the construction of hyponym ('is-a') hierarchies. A subtree in such a hierarchy can be viewed as defining a concept.

A major algorithmic approach is to represent word contexts as vectors in some space and use distributional measures and clustering in that space. Pereira (1993), Curran (2002) and Lin (1998) use syntactic features in the vector definition. (Pantel and Lin, 2002) improves on the latter by clustering by committee. Caraballo (1999) uses conjunction and appositive annotations in the vector representation. Several studies avoid requiring any syntactic annotation. Some methods are based on decomposition of a lexically-defined matrix (by SVD, PCA etc), e.g. (Schütze, 1998; Deerwester et al., 1990).

While great effort has been made for improving the computational complexity of distributional methods (Gorman and Curran, 2006), they still remain highly computationally intensive in comparison to pattern approaches (see below), and most of them do not scale well for very large datasets.

The second main approach is to use lexico-syntactic patterns. Patterns have been shown to produce more accurate results than feature vectors, at a lower computational cost on large corpora (Pantel et al., 2004). Since (Hearst, 1992), who used a manually prepared set of initial lexical patterns, numerous pattern-based methods have been proposed for the discovery of concepts from seeds. Other studies develop concept acquisition for on-demand tasks where concepts are defined by user-provided seeds. Many of these studies utilize information obtained by language-specific parsing and named entity recognition tools (Dorow et al., 2005). Pantel et al. (2004) reduce the depth of linguistic data used, but their method requires POS tagging.

TextRunner (Banko et al., 2007) utilizes a set of pattern-based seed-less strategies in order to extract relational tuples from text. However, this system contains many language-specific modules, including the utilization of a parser in one of the processing stages. Thus the majority of the existing pattern-based concept acquisition systems rely on pattern/word seeds or supervised language-specific tools, some of which are very inefficient.

Davidov and Rappoport (2006) developed a framework which discovers concepts based on high frequency words and symmetry-based pattern graph properties. This framework allows a fully unsupervised seed-less discovery of concepts without relying on language-specific tools. However, it completely ignores potentially useful syntactic or morphological information.

For example, the pattern 'X and his Y' is useful for acquiring the concept of family member types, as in "his siblings and his parents'. Without syntactic information, it can capture noise, as in "... in ireland) and his wife)" (parentheses denote syntactic constituent boundaries). As another example, the useful symmetric pattern "either X or Y" can appear in both good examples ("choose either *Chihuahua* or *Collie*.") and bad ones ("either *Collie* or *Australian* Bulldog"). In the latter case, the algorithm both captures noise ("Australlian" is now considered as a candidate for the 'dog type' concept), and misses the discovery of a valid multiword candidate ("Australlian Bulldog"). While symmetry-based filtering greatly reduces such noise, the basic problem remains. As a result, incorporating at least some parsing information in a language-independent and efficient manner could be beneficial.

Unsupervised parsing has been explored for several decades (see (Clark, 2001; Klein, 2005) for recent reviews). Recently, unsupervised parsers have for the first time outperformed the right branching heuristic baseline for English. These include CCM (Klein and Manning, 2002), the DMV and DMV+CCM models (Klein and Manning, 2004), (U)DOP based models (Bod, 2006a; Bod, 2006b; Bod, 2007), an exemplar based approach (Dennis, 2005), guiding EM using contrastive estimation (Smith and Eisner, 2006), and the incremental parser of Seginer (2007) which we use here. These works learn an unlabeled syntactic structure, dependency or constituency. In this work we use constituency trees as our syntactic representation.

Another important factor in concept acquisition is the source of textual data used. To take advantage of the rapidly expanding web, many of the proposed frameworks utilize web queries rather than local corpora (Etzioni et al., 2005; Davidov et al., 2007; Pasca and Van Durme, 2008; Davidov and Rappoport, 2009). While these methods have a definite practical advantage of dealing with the most recent and comprehensive data, web-based evaluation has some methodological drawbacks such as limited repeatability (Kilgarriff, 2007). In this study we apply our framework on offline corpora in settings similar to that of previous work, in order to be able to make proper comparisons.

3 Efficient Unsupervised Parsing

Our method utilizes the information induced by unsupervised parsers. Specifically, we make use of the bracketings induced by Seginer's parser[1] (Seginer, 2007). This parser has advantages in three major aspects relevant to this paper.

First, it achieves state of the art unsupervised parsing performance: its F-score[2] is 75.9% for sentences of up to 10 words from the PennTreebank Wall Street Journal corpus (WSJ) (Marcus, 1993), and 59% for sentences of the same length from the German NEGRA (Brants, 1997) corpus. These corpora consists of newspaper texts.

Second, to obtain good results, manually created POS tags are used as input in all the unsupervised parsers mentioned above except of Seginer's, which uses raw sentences as input. (Headden et al., 2008) have shown that the performance of algorithms that require POS tags substantially decreases when using POS tags induced by unsupervised POS taggers instead of manually created ones. Seginer's incremental parser is therefore the only *fully* unsupervised parser providing high quality parses.

Third, Seginer's parser is extremely fast. During its initial stage, the parser builds a lexicon. Our Pentium 2.8GHB machines with 4GHB RAM can store in memory the lexicon created by up to 0.2M sentences. We thus divided our corpora to batches of 0.2M sentences and parsed each of them separately. Note that in this setup parsing quality might be even better than the quality reported in (Seginer, 2007), since in the setup reported in that paper the parser was applied to a few thousand sentences only. On average, the parsing time of a single batch was 5 minutes (run time did not significantly differ across batches and corpora).

Parser description. The parser utilizes the novel common-cover link representation for syntactic structure. This representation resembles dependency structure but unlike the latter, it can be translated into a constituency tree, which is the syntactic representation we use in this work.

The parsing algorithm creates the common-cover links structure of a sentence in an incremental manner. This means that the parser reads the words of a sentence one after the other and, as each word is read, it is only allowed to add links that have one of their ends at that words (and update existing ones). Words which have not yet been read are not avail-

[1] The parser is freely available at *http://staff.science.uva.nl/∼yseginer/ccl*

[2] $F = \frac{2 \cdot R \cdot P}{R+P}$, where R and P are the recall and precision of the parsers' bracketing compared to manually created bracketing of the same text. This is the accepted measure for parsing performance (see (Klein, 2005)).

able to the parser at this stage. This restriction is inspired by psycholinguistics research which suggests that humans process language incrementally. This results in a significant restriction of the parser's search space, which is the reason it is so fast.

During its initial stage the parser builds a lexicon containing, for each word, statistics helping the decision of whether to link that word to other words. The lexicon is updated as any new sentence is read. Lexicon updating is also done in an incremental manner so this stage is also very fast.

4 Unsupervised Pattern Discovery

In the first stage of our algorithm, we run the unsupervised parser on the corpus in order to produce a bracketing structure for each sentence. In the second stage, described here, we use these bracketings in order to discover, in a fully unsupervised manner, patterns that could be useful for concept mining.

Our algorithm is based on the concept acquisition method of (Davidov and Rappoport, 2006). We discover patterns that connect terms belonging to the same concept in two main stages: discovery of pattern candidates, and identification of the symmetric patterns among the candidates.

Pattern candidates. A major idea of (Davidov and Rappoport, 2006) is that a few dozen high frequency words (HFW) such as 'and' and 'is' connect other, less frequent content terms into relationships. They define *meta-patterns*, which are short sequences of H's and C's, where H is a slot for a HFW and C is a slot for a content word (later to become a word belonging to a discovered concept). Their method was shown to produce good results. However, the fact that it does not consider any syntactic information causes problems. Specifically, it does not consider the constituent structure of the sentence. Meta-patterns that cross constituent boundaries are likely to generate noise – two content words (C's) in a meta-pattern that belong to different constituents are likely to belong to different concepts as well. In addition, meta-patterns that do not occupy a full constituent are likely to 'cut' multiword expressions (MWEs) into two parts, one part that gets treated as a valid C word and one part that is completely ignored.

The main idea in the present paper is to use the bracketings induced by unsupervised parsers in order to avoid the problems above. We utilize bracketing boundaries in our meta-patterns in addition to HFW and C slots. In other words, their original meta-patterns are totally lexical, while ours are lexico-syntactic meta-patterns. We preserve the attractive properties of meta-patterns, because both HFWs and bracketings can be found or computed in a language independent manner and very efficiently.

Concretely, we define a HFW as a word appearing more than T_H times per million words, and a C as a word or multiword expression containing up to 4 words, appearing less than T_C times per million.

We require that our patterns include two slots for C's, separated by at least a single HFW or bracket. We allow separation by a single bracket because the lowest level in the induced bracketing structure usually corresponds to lexical items, while higher levels correspond to actual syntactic constituents.

In order to avoid truncation of multiword expressions, we also require the meta pattern to start and end by a HFW or bracket. Thus our meta-patterns match the following regular expression:

$$\{H|B\}^* \; C_1 \; \{H|B\}^+ \; C_2 \; \{H|B\}^*$$

where "*" means zero or more times, and "+" means one or more time and B can be "(","")" brackets produced by the parser (in these patterns we do not need to guarantee that brackets match properly). Examples of such patterns include "$((C_1)in\ C_2))$", "$(C_1)(such(as(((C_2)$", and "$(C_1)and(C_2)$"[3]. We dismiss rare patterns that appear less than T_P times per million words.

Symmetric patterns. Many of the pattern candidates discovered in the previous stage are not usable. In order to find a usable subset, we focus on the symmetric patterns. We define a symmetric pattern as a pattern in which the same pair of terms (C words) is likely to appear in both left-to-right and right-to-left orders. In order to identify symmetric patterns, for each pattern we define a pattern graph $G(P)$, as proposed by (Widdows and Dorow, 2002). If term pair (C_1, C_2) appears in pattern P in some context,

[3] This paper does not use any punctuation since the parser is provided with sentences having all non-alphabetic characters removed. We assume word separation. $C_{1,2}$ can be a word or a multiword expression.

we add nodes c_1, c_2 to the graph and a directed edge $E_P(c_1, c_2)$ between them. In order to select symmetric patterns, we create such a pattern graph for every discovered pattern, and create a symmetric subgraph SymG(P) in which we take only bidirectional edges from $G(P)$. Then we compute three measures for each pattern candidate as proposed by (Davidov and Rappoport, 2006):

$$M_1(P) := \frac{|\{c_1|\exists c_2 E_P(c_1, c_2) \wedge \exists c_3 E_P(c_3, c_1)\}|}{|Nodes(G(P))|}$$

$$M_2(P) := \frac{|Nodes(SymG(P))|}{|Nodes(G(P))|}$$

$$M_3(P) := \frac{|Edges(SymG(P))|}{|Edges(G(P))|}$$

For each measure, we prepare a sorted list of all candidate patterns. We remove patterns that are not in the top Z_T (we use 100, see Section 6) in any of the three lists, and patterns that are in the bottom Z_B in at least one of the lists.

5 Concept Discovery

At the end of the previous stage we have a set of symmetric patterns. We now use them in order to discover concepts. The concept discovery algorithm is essentially the same as used by (Davidov and Rappoport, 2006) and has some similarity with the one used by (Widdows and Dorow, 2002). In this section we outline the algorithm.

The clique-set method. The utilized approach to concept discovery is based on connectivity structures in the all-pattern term relationship graph G, resulting from merging all of the single-pattern graphs for symmetric patterns selected in the previous stage. The main observation regarding G is that highly interconnected words are good candidates to form a concept. We find all strong n-cliques (subgraphs containing n nodes that are all interconnected in both directions). A clique Q defines a concept that contains all of the nodes in Q plus all of the nodes that are (1) at least unidirectionally connected to all nodes in Q, and (2) bidirectionally connected to at least one node in Q. Using this definition, we create a concept for each such clique.

Note that a single term can be assigned to several concepts. Thus a clique based on a connection of the word 'Sun' to 'Microsoft' can lead to a concept of computer companies, while the connection of 'Sun' to 'Earth' can lead to a concept of celestial bodies.

Reducing noise: merging and windowing. Since any given term can participate in many cliques, the algorithm creates overlapping categories, some of which redundant. In addition, due to the nature of language and the imperfection of the corpus some noise is obviously to be expected. We enhance the quality of the obtained concepts by merging them and by windowing on the corpus. We merge two concepts Q, R, iff there is more than a 50% overlap between them: $(|Q \bigcap R| > |Q|/2) \wedge (|Q \bigcap R| > |R|/2)$. In order to increase concept quality and remove concepts that are too context-specific, we use a simple corpus windowing technique. Instead of running the algorithm of this section on the whole corpus, we divide the corpus into windows of equal size and perform the concept discovery algorithm of this section (without pattern discovery) on each window independently. We now have a set of concepts for each window. For the final set, we select only those concepts that appear in at least two of the windows. This technique reduces noise at the potential cost of lowering coverage.

A decrease in the number of windows should produce more noisy results, while discovering more concepts and terms. In the next section we show that while windowing is clearly required for a large corpus, incorporation of parser data increases the quality of the extracted corpus to the point where windowing can be significantly reduced.

6 Results

In order to estimate the quality of concepts and to compare it to previous work, we have performed both automatic and human evaluation. Our basic comparison was to (Davidov and Rappoport, 2006) (we have obtained their data and utilized their algorithm), where we can estimate if incorporation of parser data can solve some fundamental weaknesses of their framework. In the following description, we call their algorithm **P** and our parser-based framework **P+**. We have also performed an indirect comparison to (Widdows and Dorow, 2002).

While there is a significant number of other related studies[4] on concept acquisition (see Section 2),

[4] Most are supervised and/or use language-specific tools.

direct or even indirect comparison to these works is problematic due to difference in corpora, problem definitions and evaluation strategies. Below we describe the corpora and parameters used in our evaluation and then show and discuss WordNet-based and Human evaluation settings and results.

Corpora. We performed in-depth evaluation in two languages, English and Russian, using three corpora, two for English and one for Russian. The first English corpus is the BNC, containing about 100M words. The second English corpus, DMOZ(Gabrilovich and Markovitch, 2005), is a web corpus obtained by crawling URLs in the Open Directory Project (dmoz.org), resulting in 68GB containing about 8.2G words from 50M web pages. The Russian corpus (Davidov and Rappoport, 2006) was assembled from web-based Russian repositories, to yield 33GB and 4G words. All of these corpora were also used by (Davidov and Rappoport, 2006) and BNC was used in similar settings by (Widdows and Dorow, 2002).

Algorithm parameters. The thresholds T_H, T_C, T_P, Z_T, Z_B, were determined mostly by practical memory size considerations: we computed thresholds that would give us the maximal number of terms, while enabling the pattern access table to reside in main memory. The resulting numbers are $100, 50, 20, 100, 100$. Corpus window size was determined by starting from a small window size, extracting at random a single window, running the algorithm, and iterating this process with increased $\times 2$ window sizes until reaching a desired vocabulary concept participation percentage (before windowing) (i.e., x% of the different words in the corpus participate in terms assigned into concepts. We used 5%.). We also ran the algorithm without windowing in order to check how well the provided parsing information can help reduce noise. Among the patterns discovered are the ubiquitous ones containing "and","or", e.g. '((X) or (a Y))', and additional ones such as 'from (X) to (Y)'.

Influence of parsing data on number of discovered concepts. Table 1 compares the concept acquisition framework with (P+) and without (P) utilization of parsing data.

	V	W		C		AS	
		P	P+	P	P+	P	P+
DMOZ	16	330	**504**	**142**	130	12.8	**16.0**
BNC	0.3	25	**42**	**9.6**	8.9	10.2	**15.6**
Russ.	10	235	**406**	**115**	96	11.6	**15.1**

Table 1: Results for concept discovery with (P+) and without (P) utilization of parsing data. V is the total number (millions) of different words in the corpus. W is the number (thousands) of words belonging to at least one of the terms for one of the concepts. C is the number (thousands) of concepts (after merging and windowing). AS is the average(words) category size.

We can see that the amount of different words covered by discovered concepts raises nearly 1.5-fold when we utilize patterns based on parsing data in comparison to pure HFW patterns used in previous work. We can also see nearly the same increase in average concept size. At the same time we observe about 15% reduction in the total number of discovered concepts.

There are two opposite factors in P+ which may influence the number of concepts, their size and coverage in comparison to P. On one hand, utilization of more restricted patterns that include parsing information leads to a reduced number of concept term instances being discovered. Thus, the P+ pattern "(X (or (a Y))" will recognize "(TV (or (a movie))" instance and will miss "(lunch) or (a snack))", while the P pattern "X or a Y" will capture both. This leads to a decrease in the number of discovered concepts.

On the other hand, P+ patterns, unlike P ones, allow the extraction of multiword expressions[5], and indeed more than third of the discovered terms using P+ were MWEs. Utilization of MWEs not only allows to cover a greater amount of different words, but also increases the number of discovered concepts since new concepts can be found using cliques of newly discovered MWEs. From the results, we can see that for a given concept size and word coverage, the ability to discover MWEs overcomes the disadvantage of ignoring potentially useful concepts.

Human judgment evaluation. Our human judgement evaluation closely followed the protocol (Davidov and Rappoport, 2006).

We used 4 subjects for evaluation of the English

[5] While P method can potentially be used to extract MWEs, preliminary experimentation shows that without significant modification, quality of MWEs obtained by P is very low in comparison to P+

concepts and 4 subjects for Russian ones. In order to assess subjects' reliability, we also included random concepts (see below). The goal of the experiment was to examine the differences between the P+ and P concept acquisition frameworks. Subjects were given 50 triplets of words and were asked to rank them using the following scale: (1) the words definitely share a significant part of their meaning; (2) the words have a shared meaning but only in some context; (3) the words have a shared meaning only under a very unusual context/situation; (4) the words do not share any meaning; (5) I am not familiar enough with some/all of the words.

The 50 triplets were obtained as follows. We have randomly selected 40 concept pairs (C+,C): C+ in P+ and C in P using five following restrictions: (1) concepts should contain at least 10 words; (2) for a selected pair, C+ should share at least half of its single-word terms with C, and C should share at least half of its words with C+; (3) C+ should contain at least 3 MWEs; (4) C should contain at least 3 words not appearing in C+; (5) C+ should contain at least 3 single-word terms not appearing in C.

These restrictions allow to select concept pairs such that C+ is similar to C while they still carry enough differences which can be examined. We selected the triplets as following: for pairs (C+, C) ten triplets include terms appearing in both C+ and C (*Both* column in Table 2), ten triplets include single-word terms appearing in C+ but not C (*P+ single* column), ten triplets include single-word terms appearing in C but not C+ (*P* column), ten triplets include MWEs appearing in C+ (*P+ mwe* column) and ten triplets include random terms obtained from P+ concepts (*Rand* column).

	P+		P	Both	Rand
	mwe	single			
% shared meaning					
DMOZ	85	**88**	68	81	6
BNC	85	**90**	61	88	0
Russ.	89	**95**	70	93	11
triplet score (1-4)					
DMOZ	1.7	**1.4**	2.5	1.7	3.8
BNC	1.6	**1.3**	2.1	1.5	4.0
Russ.	1.5	**1.1**	2.0	1.3	3.7

Table 2: Results of evaluation by human judgment of three data sets. P+ single/mwe: single-word/MWE terms existing only in P+ concept; P: single-word terms existing only in P concept; Both: terms existing in both concepts; Rand: random terms. See text for detailed explanations.

The first part of Table 2 gives the average percentage of triplets that were given scores of 1 or 2 (that is, 'significant shared meaning'). The second part gives the average score of a triplet (1 is best). In these lines scores of 5 were not counted. Inter-evaluator Kappa between scores are 0.68/0.75/0.76 for DMOZ, BNC and Russian respectively. We can see that terms selected by P and skipped by P+ receive low scores, at the same time even single-word terms selected by P+ and skipped by P show very high scores. This shows that using parser data, the proposed framework can successfully avoid selection of erroneous terms, while discovering high-quality terms missed by P. We can also see that P+ performance on MWEs, while being slightly inferior to the one for single-word terms, still achieves results comparable to those of single-word terms.

Thus our algorithm can greatly improve the results not only by discovering of MWEs but also by improving the set of single word concept terms.

WordNet-based evaluation. The major guideline in this part of the evaluation was to compare our results with previous work (Davidov and Rappoport, 2006; Widdows and Dorow, 2002) without the possible bias of human evaluation. We have followed their methodology as best as we could, using the same WordNet (WN) categories and the same corpora. This also allows indirect comparison to several other studies, thus (Widdows and Dorow, 2002) reports results for an LSA-based clustering algorithm that are vastly inferior to the pattern-based ones.

The evaluation method is as follows. We took the exact 10 WN subsets referred to as 'subjects' in (Widdows and Dorow, 2002), and removed all multi-word items. We then selected at random 10 pairs of words from each subject. For each pair, we found the largest of our discovered concepts containing it. The various morphological forms or clear typos of the same word were treated as one in the evaluation.

We have improved the evaluation framework for Russian by using the Russian WordNet (Gelfenbeynand et al., 2003) instead of back-translations as done in (Davidov and Rappoport, 2006). Preliminary examination shows that this has no apparent effect on the results.

For each found concept C containing N words, we computed the following: (1) Precision: the number of words present in both C and WN divided by N; (2) Precision*: the number of correct words divided by N. Correct words are either words that appear in the WN subtree, or words whose entry in the American Heritage Dictionary or the Britannica directly defines them as belonging to the given class (e.g., 'murder' is defined as 'a crime'). This was done in order to overcome the relative poorness of WN; (3) Recall: the number of words present in both C and WN divided by the number of words in WN; (4) The percentage of correctly discovered words (according to Precision*) that are not in WN.

Table 3 compares the macro-average of these 10 categories to corresponding related work. We do not observe apparent rise in precision when comparing P+ and P, but we can see significant improvement in both recall and precision* for all of three corpora. In combination with human judgement results, this suggests that the P+ framework successfully discovers more correct terms not present in WN. This causes precision to remain constant while precision* improves significantly. Rise in recall also shows that the P+ framework can discover significantly more correct terms from the same data.

	Prec.	Prec.*	Rec.	%New
DMOZ				
P	**79.8**	86.5	22.7	2.5
P+	79.5	**91.3**	**28.6**	**3.7**
BNC				
P	92.76	95.72	7.22	0.4
P+	**93.0**	**96.1**	**14.6**	**1.7**
Widdows	82.0	-	-	-
Russian				
P	82.39	89.64	20.03	2.1
P+	**83.5**	**92.6**	**29.6**	**4.0**

Table 3: WordNet evaluation in comparison to P (Davidov and Rappoport, 2006) and to Widdows(Widdows and Dorow, 2002). Columns show average precision, precision* (as defined in text), recall, and % of new words added to corresponding WN subtree.

Windowing requirement. As discussed in Section 5, windowing is required for successful noise reduction. However, due to the increase in pattern quality with parser data, it is likely that less noise will be captured by the discovered patterns. Hence, windowing could be relaxed allowing to obtain more data with sufficiently high precision.

In order to test this issue we applied our algorithms on the DMOZ corpus with 3 different windowing settings: (1) choosing window size as described above; (2) using $\times 4$ larger window; (3) avoiding windowing altogether. Each time we randomly sampled a set of 100 concepts and tagged (by the authors) noisy ones. A concept is considered to be noisy if it has at least 3 words unrelated to each other. Table 4 shows results of this test.

	Reg. Window	$\times 4$ Window	No windowing
P	4	18	33
P+	4	5	21

Table 4: Percentage of noisy concepts as a function of windowing.

We can see that while windowing is still essential even with available parser data, using this data we can significantly reduce windowing requirements, allowing us to discover more concepts from the same data.

Timing requirements are modest, considering we parsed such large amounts of data. BNC parsing took 45 minutes, and the total single-machine processing time for the 68Gb DMOZ corpus was 4 days[6]. In comparison, a state-of-art supervised parser (Charniak and Johnson, 2005) would process the same amount of data in 1.3 years[7].

7 Discussion

We have presented a framework which utilizes an efficient fully unsupervised parser for unsupervised pattern-based discovery of concepts. We showed that utilization of unsupervised parser in pattern acquisition not only allows successful extraction of MWEs but also improves the quality of obtained concepts, avoiding noise and adding new terms missed by the parse-less approach. At the same time, the framework remains fully unsupervised, allowing its straightforward application to different languages as supported by our bilingual evaluation.

This research presents one more step towards the merging of fully unsupervised techniques for lexical acquisition, allowing to extract semantic data without strong assumptions on domain or language. While we have aimed for concept acquisition, the proposed framework can be also useful for extraction of different types of lexical relationships, both among concepts and between concept terms.

[6] In fact, we used a PC cluster, and all 3 corpora were parsed in 15 hours.

[7] Considering the reported parsing rate of 10 sentences per second

References

Mishele Banko, Michael J Cafarella , Stephen Soderland, Matt Broadhead, Oren Etzioni, 2007. Open Information Extraction from the Web. *IJCAI '07*.

Rens Bod, 2006a. An All-Subtrees Approach to Unsupervised Parsing. *ACL '06*.

Rens Bod, 2006b. Unsupervised Parsing with U-DOP. *CoNLL X*.

Rens Bod, 2007. Is the End of Supervised Parsing in Sight? *ACL '07*.

Thorsten Brants, 1997. The NEGRA Export Format. *CLAUS Report, Saarland University*.

Sharon Caraballo, 1999. Automatic Construction of a Hypernym-labeled Noun Hierarchy from Text. *ACL '99*.

Eugene Charniak and Mark Johnson, 2005. Coarse-to-Fine n-Best Parsing and MaxEnt Discriminative Reranking. *ACL '05*.

Alexander Clark, 2001. *Unsupervised Language Acquisition: Theory and Practice*. Ph.D. thesis, University of Sussex.

James R. Curran, Marc Moens, 2002. Improvements in Automatic Thesaurus Extraction *SIGLEX 02'*, 59–66.

Dmitry Davidov, Ari Rappoport, 2006. Efficient Unsupervised Discovery of Word Categories using Symmetric Patterns and High Frequency Words. *COLING-ACL '06*.

Dmitry Davidov, Ari Rappoport, Moshe Koppel, 2007. Fully Unsupervised Discovery of Concept-Specific Relationships by Web Mining. *ACL '07*.

Dmitry Davidov, Ari Rappoport, 2009. Translation and Extension of Concepts Across Languages. *EACL '09*.

Scott Deerwester, Susan Dumais, George Furnas, Thomas Landauer, Richard Harshman, 1990. Indexing by Latent Semantic Analysis. *J. of the American Society for Info. Science*, 41(6):391–407.

Simon Dennis, 2005. An exemplar-based approach to unsupervised parsing. *Proceedings of the 27th Conference of the Cognitive Science Society*.

Beate Dorow, Dominic Widdows, Katarina Ling, Jean-Pierre Eckmann, Danilo Sergi, Elisha Moses, 2005. Using curvature and Markov clustering in Graphs for Lexical Acquisition and Word Sense Discrimination. *MEANING '05*.

Oren Etzioni, Michael Cafarella, Doug Downey, S. Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel Weld, Alexander Yates, 2005. Unsupervised Named-entity Extraction from the Web: An Experimental Study. *Artificial Intelligence, 165(1):91134*.

Evgeniy Gabrilovich, Shaul Markovitch, 2005. Feature Generation for Text Categorization Using World Knowledge. *IJCAI '05*.

Ilya Gelfenbeyn, Artem Goncharuk, Vladislav Lehelt, Anton Lipatov, Victor Shilo, 2003. Automatic Translation of WordNet Semantic Network to Russian Language (in Russian) *International Dialog 2003 Workshop*.

James Gorman, James R. Curran, 2006. Scaling Distributional Similarity to Large Corpora. *COLING-ACL '06*.

William P. Headden III, David McClosky and Eugene Charniak, 2008. Evaluating Unsupervised Part-of-Speech tagging for Grammar Induction. *COLING '08*.

Marti Hearst, 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. *COLING '92*.

Adam Kilgarriff, 2007. Googleology is Bad Science. *Computational Linguistics '08, Vol.33 No. 1,pp147-151.* .

Dan Klein and Christopher Manning, 2002. A generative constituent-context model for improved grammar induction. *Proc. of the 40th Meeting of the ACL*.

Dan Klein and Christopher Manning, 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. *ACL '04*.

Dan Klein, 2005. *The unsupervised learning of natural language structure.* Ph.D. thesis, Stanford University.

Hang Li, Naoki Abe, 1996. Clustering Words with the MDL Principle. *COLING '96*.

Dekang Lin, 1998. Automatic Retrieval and Clustering of Similar Words. *COLING '98*.

Marcus Mitchell P., Beatrice Santorini and Mary Ann Marcinkiewicz, 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330

Marius Pasca, Benjamin Van Durme, 2008. Weakly-supervised Acquisition of Open-domain Classes and Class Attributes from Web Documents and Query Logs. *ACL 08*.

Patrick Pantel, Dekang Lin, 2002. Discovering Word Senses from Text. *SIGKDD '02*.

Patrick Pantel, Deepak Ravichandran, Eduard Hovy, 2004. Towards Terascale Knowledge Acquisition. *COLING '04*.

Fernando Pereira, Naftali Tishby, Lillian Lee, 1993. Distributional Clustering of English Words. *ACL '93*.

Hinrich Schütze, 1998. Automatic Word Sense Discrimination. *Computational Linguistics* , 24(1):97–123.

Yoav Seginer, 2007. Fast Unsupervised Incremental Parsing. *ACL '07*.

Noah A. Smith and Jason Eisner, 2006. Annealing Structural Bias in Multilingual Weighted Grammar Induction . *ACL '06*.

Dominic Widdows, Beate Dorow, 2002. A Graph Model for Unsupervised Lexical Acquisition. *COLING '02*.