Learning Corpus Patterns Using Finite State Automata

Octavian Popescu
FBK-irst, Trento, Italy
popescu@fbk.eu

1 Introduction

Words get their meaning in context and Harris's Distributional Hypothesis has been used in computational linguistics in order to identify the relationship between co-occurring words and their senses. In general, the local context contains the necessary information for word sense disambiguation (Stevenson&Wilks 2001). However, the exact extent of the local context varies significantly. To cope with this problem, previous research has shown that the regularity of word usage in natural language can be exploited (Pustejovsky&Hanks 2001). Many times, words are used in phrases with a patternable structure. On the basis of corpus evidence (Popescu&Magnini 2007), or on the basis of the lexicographer's intuition on the normal usage (Hanks 2005) a set of patterns can be built which makes the link between context and word senses.

In this paper [1] we focus on patterns centered on verbs. We show that their structure is learnable and by employing a learning algorithm we are able to build a recognizer able to match such patterns against previously unseen text. The CPA resource (Hanks & Pustejovsky 2005, Pustejovsky & Jezek 2008) contains a set of patterns for a part of the English verbs and is built through a systematic analysis of the patterns of meaning and use for each verb. Meaning is associated with prototypical sentences which are extracted from the BNC. The slots of the patterns are specified with semantic types. For example, the sentences:

(ACP) ... least that intense moment before the body abandons itself to passion.
(CCN) They danced wildly down the street, abandoning themselves to the night and the moon.

are instances of the pattern:

HUMAN abandon SELF {to ACTIVITY | to ATTITUDE}

HUMAN, SELF etc. are semantic types. The use of {} signals an optional slot of the pattern and | signals a choice. A semantic type characterizes a whole class of nouns, and as such, the semantic types are organized in a shallow ontology. The structure of these patterns is regular and we show that we can use the Angluin Algorithm to build a finite state automaton (FSA) which can recognize the patterns.

Going from the set of sentences associated to each pattern to the FSA recognizer is not trivial. The CPA does not contain information regarding the syntax of the patterns, or the senses of the words inside a pattern and it does not provide a resource which assigns a list of possible semantic types to the nouns of the English language. In order to obtain this information, we must rely on parsing and on other two resources, WordNet(Miller) and SUMO(Niles&Pease 2001). WordNet is a sense repository and SUMO is an ontology aligned to WordNet senses. We use SUMO to associate semantic types to the nouns. In the training phase, which results in the construction of the FSA recognizer, the system learns how to identify a certain pattern in a text where the words are replaced with SUMO semantic types. By matching a pattern, we obtain the syntactic structure of the context and the senses of the words in the context due to the SUMO alignment to WordNet. In the experiments we ran, we tested both the accuracy in finding the

[1] This research is supported by the BCROCE project. The author also thanks Nam Khanh Tran for helping implementing the Angluin Algorithm

correct syntactic structure and the accuracy in predicting the correct sense of the words of the matched context.

We introduce the task of pattern matching. Given an arbitrary sentence for which we know there is a unique pattern that matches it, the task consists in finding the appropriate pattern which matches the right words in the sentence. We analyzed the performances obtained by a baseline against a SVM approach and against the FSA recognizer. The results show that both the SVM and the FSA recognizer are over the baseline by several tens of percentages. The FSA recognizer reaches a significantly better accuracy than the SVM approach. We test the approaches both by a cross validation technique and by analyzing individually the performances on a list of verbs.

This paper is organized as follow: in the next Section we review the relevant literature on the interaction between meaning, syntax, ontology and patterns. In Section 3 we describe the form of corpus patterns and the CPA resource. in Section 4 we present the way in which the Angluin Algorithm for learning regular grammars from examples can be modified to learn to recognize the corpus patterns. In Section 5 the results of the experiments we carried out are presented and discussed. In the last section we present the conclusion and further research.

2 Related Work

Based on Harris Distributional Hypothesis, many approaches to WSD have focused on the contexts formed by the words surrounding the target word. With respect to verb behaviour, selectional restrictions have been used in WSD (see among others Resnik 1997, McCarthy, Caroll, Preis 2001, Briscoe et al. 2006). Also, (Hindle 1990) has tried to classify English nouns in similarity classes by using a mutual information measure with respect to the subject and object roles. Such information is very useful only in certain cases and, as such, it is difficult to use it directly in doing WSD.

Lin and Pantel (Lin, Pantel 2001) transpose the HDH from words to dependency trees. However, their measure of similarity is based on a frequency measure. They maintain that a (slotX, he) is less indicative than a (slotX, sheriff). While this might be true in some cases, the measure of similarity is given by the behaviour of the other components of the contexts: both he and $sheriff$ act either exactly the same with respect to certain verb meanings, or totally differently with respect to others. However, their method cannot be extended to take into account such differences. A classification of these cases is instrumental for WSD. Equally important is overcoming the limitation of considering only the subject and object. It has been shown that closed class categories, especially prepositions and particles, play an important role in disambiguation and wrong predictions are made if they are not taken into account (see, among others, Collins and Brooks 1995, Stetina&Nagao 1997). Our approach addresses both these issues.

Zhao, Meyers and Grishman (Zhao, Meyers and Grishman 2004) proposed a SVM application to slot detection, which combines two different kernels, one of them being defined on dependency trees. Their method tries to identify the possible fillers for an event, but it does not attempt to treat ambiguous cases; also, the matching score algorithm makes no distinction between the importance of the words, considering equal matching score for any word within two levels of the dependency tree.

(Pederson et al. 1997-2005) have clustered together the examples that represent similar contexts for WSD. However, given that they adopt mainly the methodology of ordered pairs of bigrams of substantive words, their technique works only at the word level, which may lead to a data sparseness problem. Ignoring syntactic clues may increase the level of noise, as there is no control over the relevance of a bigram. Many of the purely syntactic methods have considered the properties of the subcategorization frame of verbs. Verbs have been partitioned in semantic classes mainly on the basis of Levins classes of alternation. (Dorr&Jones 1996, Dang et al. 1998, Collins 1989, McCarthy 2001, Korhonen 2002, Lapata Brew 2004). These semantic classes can be used in WSD via a process of alignment with hierarchies of concepts as defined in sense repository resources (Shin&Mihalcea 2005). However the problem of the consistency of alignment is still an open issue and further research must be pursued before applying these methods to WSD.

The relationship between events and dependency parsing is analyzed in (McClosky et al. 2011). They extract events at the sentence granularity. However, the fact that the senses of the words are related in describing an event is not discussed. A semi-supervised technique for the discovery of semantic pattern is presented in (Sun&Grishman 2011). Their paper takes into account only the ACE named entities - PERSON, GPE, LOCATION etc. While the authors tried to catch meaning relations between their patterns, there is no clear meaning associated with each pattern. In fact, many times different senses are found in identically syntactic contexts. To capture the differences, the semantic types must be taken into account as well. The semantic binary relations discoverable in text are the focus of the paper (Chan&Roth 2011). They individuate syntactico-semantic structures which could be encoded as patterns but they do not discuss the complexity of learning them. The paper does not discuss possible extensions of the presented method to patterns matching a whole sentence.

3 Corpus Pattern Analysis

In CPA a pattern is understood as a corpus-derived predicate-argument structure with specification of the expected semantic type and subcategorization properties for the arguments (HanksPustejovski 2005). A pattern may not include, and usually it does not, all the phrases presented into the subcategorization frame. A pattern corresponds to a subgraph of the dependency graph of a set of sentences. In Table 1, in the first column, we present three patterns of the verb abandon, and in the second column we show prototypical examples.

Patterns	*Protypical examples*
HUMAN | INSTITUTION	he abandoned plans of working
abandon	are incapacitated or have abandoned their practices
ACTIVITY | PLAN	We should not abandon the search
HUMAN | INSTITUTION	he had abandoned immediate hopes
abandon	abandoned their principles
ATTITUDE	he had abandoned his commitment to persuasion
HUMAN | GROUP	citizens of Phocaea abandoned their town
abandon	The lands that they abandoned
LOCATION	before abandoning the site

Table 1: Patterns and Prototypical Examples

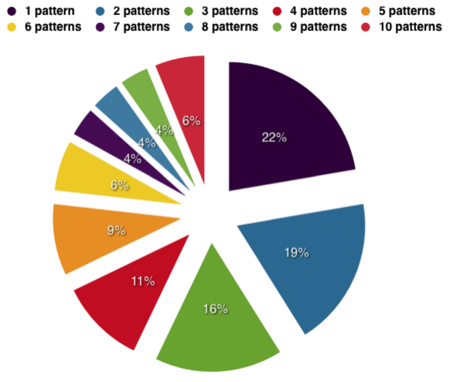
A semantic type outside a pattern is not functional. A word may be characterized by many semantic types, but only one of them is actuated in a pattern. The lexicologists task in CPA is to find the appropriate level of generalization of the semantic types on the basis of which senses are distinguished. The words collocating on the same syntactic position are grouped together according to their influence on the verb. Different patterns are often meaning contrastive. However, this is not always the case. Consider, for example, the three sentences below:

ex1 I drove him to the house.
ex2 I drove him to his father.
ex3 I drove him to despair.

which have the following corresponding patterns:

ex1pattern HUMAN drive_3 HUMAN to BUILDING
ex2pattern HUMAN drive_3 HUMAN to HUMAN
ex3pattern HUMAN drive_5 HUMAN to PSYCHOLOGICAL_STATE

Figure 1: Distribution of number of patterns



The patterns ex1pattern and ex2pattern, as opposed to ex3pattern, are not meaning contrastive. It would be hard to imagine that the same semantic type could cover both $house$ and $father$. Rather, these remain separate patterns. However, the intuition is that in ex1 and in ex2, $house$ and $father$ are both understood as PLACE. The CPA treats such cases as "exploitation of the norm" (Hanks 2008). The CPA provides a different set of sentence contexts from BNC for exploitation cases. The CPA resource is freely available from http://deb.fi.muni.cz/pdev/. Table 2 summarizes the figures related to the actual coverage of the corpus. The number of patterns varies from 1 to 56.

Characteristics	*Dimension*
Number of Verbs	721
Number of Patterns	2745
Number of files with Examples	5447

Table 2: CPA corpus in Figures

Figure 1 shows the distributions of the number of patterns in CPA. There are roughly a couple of semantic types currently used in CPA. Two of them, namely "Human" and "Institution" are significantly more frequent than others; they are used 1,849 and 365 times, respectively. The CPA also provides the likelihood of a pattern in BNC. The distribution of the patterns in corpus is not uniform, the mode being that a dominant pattern is likely to have a few times more occurrences than the next most frequent pattern.

We computed how many times the dominant pattern for a verb has more than 40%, 60% or 80% of occurrences, by also considering the total number of patterns for the respective verbs grouped in intervals: verbs which have between 3 and 5 patterns, verbs which have between 5 and 20 patterns, verbs having between 20 and 40 patterns, and verbs having between 40 and 60 patterns. For example, 65.25% of the verbs with patterns between 5 and 20 have a dominant pattern that occurs more than 40% in the corpus, but only 23.72% of the verbs with the same number of patterns have a dominant pattern that occurs more than 60% of the time in the corpus. See Table 3.

coverage/patterns	*2-5*	*6-20*	*21-40*
40%	94.35%	65.25%	25%
60%	60.45%	23.72%	12.5%
80%	27.1%	14.23%	0%

Table 3: Dominant Pattern Frequency in Corpus

The SUMO ontology is aligned to the senses present in Wordnet1.6. In Table 4 we list the SUMO attributes for the direct object position for the examples listed in Table 2.

Considering all SUMO attributes of a word is likely to lead to confusion, for example in Table 4 the "NormativeAttribute" belongs both to practice and principle, which are the direct objects in different pat-

direct object	*SUMO attributes*
plan	Plan, Abstract, icon
practice	normativeAttribute, EducationalProcess
search	Pursuing, Investigating, ContentDevelopment
hope	EmotionalState, Reasoning
principle	NormativeAttribute, Proposition
commitment	TraitAttribute, Declaring
town	City, Geopolitical
land	LandArea, Geopolitical, Nation
site	LandArea, Located

Table 4: Patterns and Prototypical Examples

terns. However, the sense determination relationship characterizing the CPA patterns (explained below) , allows only a certain combination of senses, to which only certain SUMO attributes correspond, because SUMO is aligned to the sense repository. The pattern learning and recognizing algorithm must be able to retain for a word only the SUMO features which are instantiated in a particular corpus sentence. The algorithm presented in the next section learns the patterns, as well as which SUMO attributes are legible in a CPA pattern for each word.

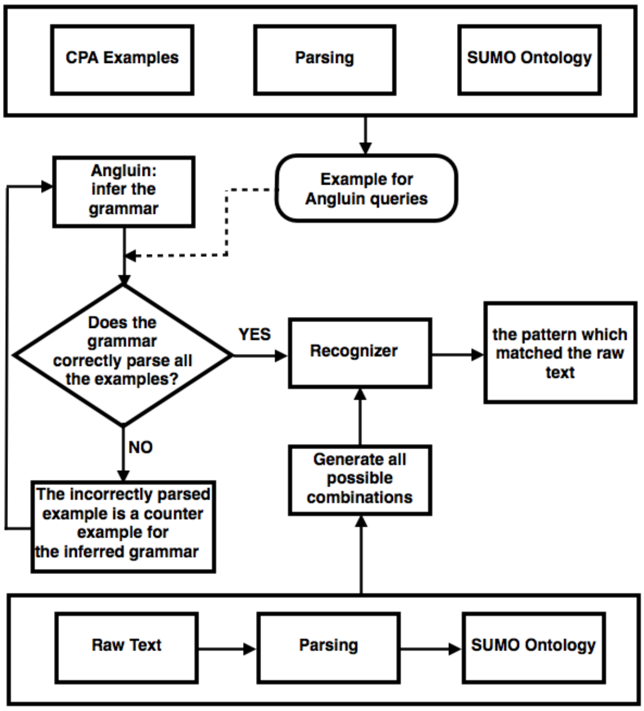
Before concluding this section we discuss a relationship between the components of the corpus patterns which will be proven to be important for the construction of more accurate FSA. The relationships between the semantic types and the senses of the verbs are such that only certain combinations are valid. We are interested in corpus patterns for which a determination relationship holds: given either the sense of the verb or the semantic types of one of the components then all the other can be inferred. For example knowing that the direct object has the semantic type LAND then the verb *abandon* must have the sense 3. The disambiguation of the senses of the words matched by a pattern follow a chain like relationship - it is enough to disambiguate one component, and all the words get disambiguated. We call this relationship Chain Clarifying Relationship (CCR) (Popescu,Magnini 2007, Popescu 2012). CCR is instrumental in constructing accurate FSAs. By considering the difference between two CCRs we do not need to match the whole pattern, but to identify only the distinctive semantic types in the CCRs. In Section Experiments we analyze the influence of this relationship on the overall accuracy of the recognizer.

4 Angluin Algorithm

The Angluin's algorithm (AA) is proved to be able to learn the minimal regular grammar that produces or rejects a set of examples provided as input. In general, the problem of learning a regular grammar only from positive examples is an NP-hard problem. Angluin's algorithm is guided in learning by an oracle, which can answer yes/no questions or give a counter example, and it runs in linear time by considering the length of the input examples.

The AA exploits the fact that a language is regular if and only if it is prefix closed, which means that a language is regular if and only if there is a finite number of equivalence classes of the strings, prefixes, which affect the acceptability of the bigger strings that they initiate in the same way. As it learns new examples, the AA builds a table of observation of all possible prefixes and suffixes. When the acceptance of each of the strings formed by joining prefixes with suffixes is known, the table is considered closed. If a closed table also obeys the prefix closeness condition, then it is also considered consistent. The entries in a closed and consistent table describe a Finite State Automaton (FSA), which correctly accepts or rejects the examples given. However, there is more that one possible regular language that describes a set of finite examples. Therefore, when the table is closed and consistent the algorithm asks for a counter

Figure 2: System Flow


CPA Examples
Parsing
SUMO Ontology
Angluin: infer the grammar
Example for Angluin queries
Does the grammar correctly parse all the examples?
YES
Recognizer
the pattern which matched the raw text
NO
Generate all possible combinations
The incorrectly parsed example is a counter example for the inferred grammar
Raw Text
Parsing
SUMO Ontology

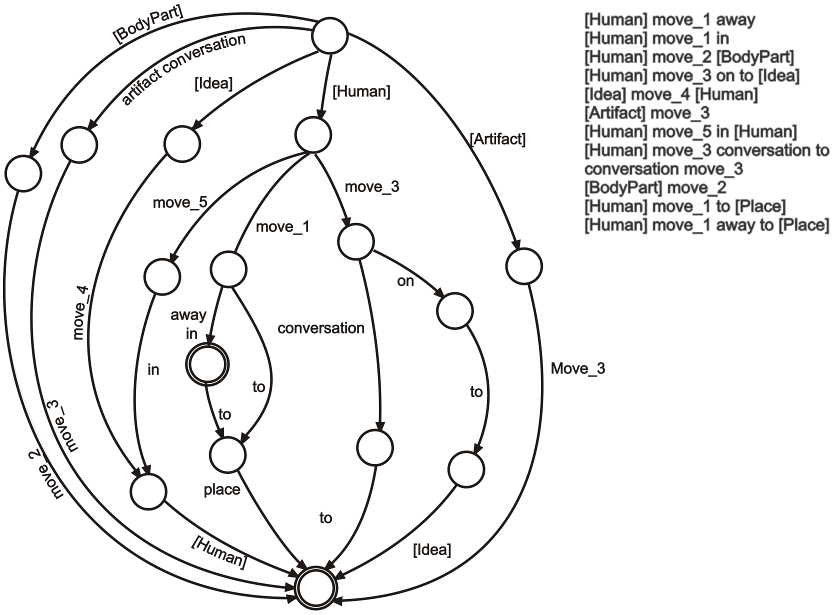
example - which is a string that is accepted or rejected by the language to be learned, and is rejected or accepted, respectively, by the language described in the actual table. If such a counter example is given, the operation of closing the table is carried out again; otherwise, the algorithm stops and the desired FSA is the one described in the table. (Angluin 1987).

The oracles questions about the acceptance of a new string formed by prefixing and suffixing parts of the previously seen strings are called the membership queries. The oracles questions regarding the equivalence between the FSA found by consistently closing the table and the FSA parsing the real grammar are called equivalence queries. A counter example to the equivalence query shows that the actual FSA is too general and new states must be found.

The AA receives as input all the strings created by considering all the SUMO attributes for the words in training and learns the correct prefixes and suffixes for the patterns. The membership queries are carried out in order to determine what SUMO attributes form valid strings in which slot. The equivalent queries are carried out to determine that no relevant SUMO attribute is left unanalyzed. If a word has many SUMO features, it generates more symbols: $practice$, for example, generates $EDUCATIONAL\_PROCESS$, and $NORMATIVE\_ATTRIBUTE$. The same string may be generated by two sentences with different patterns, for example *abandon practice* and *abandon principle*, which generates *HUMAN abandon NORMATIVE_ATTRIBUTE* (see Table 1 and Table 4). This is incorrect, because if the FSA accepts *HUMAN abandon NORMATIVE_ATTRIBUTE* then the FSA is unable to assign a unique pattern to the text. Such strings are considered counter examples for the AA algorithm and the system learns that they are not part of the language to be learned. Consequently, the respective SUMO feature for that particular slot will not be considered by the final FSA. Whether a SUMO attribute is considered or not depends entirely on the structure of the patterns for that verb. The flow is plotted in Figure 2.

The grammars we are interested in are finite. The role of the oracle can be skipped in this case. In an input file we provide the set of strings with the specification of their acceptances. The AA reads the examples from the input file and builds the table. The answer to both membership queries and equivalence queries is carried out automatically by assuming that if a string is not in the input file, then it is not accepted, and by assuming that if there are strings in the input file which are not generated by the

Figure 3: the FSA recognizer for a subset of examples for *move*



current FSA, then any of them can serve as a counter-example and the search for a new FSA resumes.

The input to the AA algorithm is a set of examples of patterns and the output is a FSA able to recognize only the strings that corresponds to the contexts which are matched by only one pattern. In Figure 3 we present the FSA generated by a subset of examples for *move*.

5 Experiments

We ran several experiments in order to evaluate the performances of pattern recognition via regular grammars. We started by running a 4 fold cross validation experiment. Because we wanted to analyze the results in more detail, we look for a set of verbs having a representative number of patterns and of examples for the whole set of verbs and we analyzed specifically the accuracy of various methods individually. The recognizing process using FSA can be made in two scenarios: using a parser or not. The second scenario, no parsing for the input text, is challenging, because the recognizer acts as syntactico-semantic parser which outputs a dependency path corresponding to the context matched and it also outputs the senses of the words . While the accuracy of pattern recognition is lower in this case, the results are promising.

The SUMO features are obtained for the noun phrases heads via a public available API (Pianta et al. 2002). At the test phase all the possible SUMO combinations inside the syntactic slots of a pattern are given to FSA. If the FSA is unable to find a derivation, or if it finds more than one, it means that we are unable to match a single pattern against the given sentence and these cases are considered errors.

The results for the 4 fold cross validation experiment are presented in Table 8. Both the SVM and the FSA reaches a good accuracy. However, the results may be biased by the existence of verbs having just one pattern or of verbs having a dominant pattern. In such cases, which represents more or less a quart of the total number, there is no ambiguity so we can hardly talk about a recognition process. For a clearer understanding of the behavior of the systems we chose a set of 12 verbs having a number of patterns between 3 and 9, half of them having exactly 5 patterns (see Table 5). The maximal and the minimal frequencies of a pattern are listed in the third and forth column, respectively.

We are interested in the maximal and minimal frequencies of the pattern, because, usually, there is little training available for those patterns with low frequency. The risk of not recognizing the minimal frequency is high. The approach presented here depends to a little extent on the dimension of the training corpus and to a large extent on its quality. That is why we wanted to analyze the performances for

verb	*pattern*	*max Freq*	*min Freq*	*# train 10%*
abandon	8	48%	1%	41
accompany	5	31%	1%	23
acknowledge	5	54%	1%	56
acquire	5	51%	2%	46
arrive	5	69%	1%	41
execute	5	36%	8%	60
fence	3	64%	2%	5
furnish	4	31%	14%	21
launch	6	60%	3%	41
maintain	5	67%	2%	9
saddle	4	71%	2%	9
yield	9	24%	4%	55

Table 5: Test Verbs

different types of patterns. The available sentences were divided randomly into training and test sets. We considered approximately two training sets containing approximately 10% and 30% of all the available sentences, respectively. With a training ratio of 10%, 8 verbs had between 40 and 50 sentences. Two verbs, $accompany$ and $furnish$, have around 20 examples each, and two other verbs have only 5 and, respectively, 9 examples each (see column 5 Table 6). The 30% training sets had three times more examples. The very first run we tried was to use all SUMO features, which led to the acceptance of all the possible combinations. The result was very low; in more than 90 percent of the cases when the recognition set was not empty, it contained more than a pattern. This experiment showed the necessity of observing the CCR condition for the CPA patterns. If the CCR condition is observed, then not all the SUMO attribute combinations are accepted. All the following experiments are conducted by observing the CCR condition (see section 4). Using a 10% ratio for training was enough to obtain a very good precision, on average between 80% and 90%. However, $fence$ expectedly performed poorer than the rest, with a precision of 45%, as it contained only 5 training examples. Considering the precision for two other verbs with a relatively low number of training examples, namely $accompany$ and $furnish$, we can see that 20 examples seem to be enough for a precision around 96% (Table 7).

The low figure for recall has three main different causes: (1) the errors along the pipe generated at parsing time and at dependency extraction (2) the lack of SUMO features for pronouns and proper names and (3) the rigid condition of recognizing all the elements of a pattern, as requested by the FSA.

verb	*BasicFSA 10% train RECALL*	*ExtendedFSA 10% train RECALL*
abandon	.26	.36
accompany	.22	.49
acknowledge	.10	.12
acquire	.25	.48
arrive	.25	.37
execute	.10	.22
fence	.23	.23
furnish	.10	.32
launch	.2	.45
maintain	.1	.36
saddle	.22	.34
yield	.14	.4

Table 6: Recall for BasicFSA vs. ExtendedFSA with 10%

The first two causes are not directly linked to the methodology described here. These causes could be addressed in an independent manner. However, the third cause is directly linked to the way the

FSA works and we wanted to focus on it. When the string corresponding to a test sentence is not complete, the FSA rejects it. As many of the patterns may differ due to the direct object or due to the prepositional complement, it suffices to correctly recognize that part of the string in order to correctly categorize the test sentence as belonging to one group or another. These subparts of the patterns can be automatically generated by comparing the patterns against each other. We can include them in the training set as well. In a second experiment we provided to the AA the automatically generated subparts of the patterns. We refer to the new automaton as extended FSA in order to distinguish it from the initial FSA trained on complete patterns, which we called BasicFSA. The recall increased significantly by using the extendedFSA. For certain verbs the recall was doubled or nearly doubled. In Table 6 the results obtained are listed. We also ran the Extended FSA with a 30% training corpus. The results are listed in Table 7.

Basic+10%				Extended+30%		
verb	*precision*	*recall*	*F1*	*precision*	*recall*	*F1*
abandon	.95	.26	.41	.97	.6	.74
accompany	.96	.22	.35	.87	.71	.78
acknowledge	.88	.10	.18	.9	.25	.39
acquire	.98	.25	.39	.97	.6	.74
arrive	.60	.25	.35	1	.41	.58
execute	.78	.10	.15	.85	.46	.59
fence	.45	.23	.30	.57	.36	.44
furnish	1	.1	.16	.84	.42	.56
launch	.99	.20	.33	.95	.79	.87
maintain	.93	.10	.17	.9	.48	.63
saddle	1	.22	.36	1	.36	.68
yield	.96	.14	.24	.96	.51	.62

Table 7: BasicFSA + 10% vs. ExtendedFSA + 30% training set

Considering a training corpus which represents 30% of the total number of corpus sentences does not mean that the training was three times more informative than a 10% training corpus. This happens because it is not unusual for otherwise different sentences to have the same word on the same spot in the argument structure. If two such sentences were in the training set, there was nothing new to learn. It seemed that the precision is not affected by the dimension of the training set. We noticed that even the low frequency patterns were correctly identified. However, the increase in recall is significant. Both the increasing of the training set and the improvement brought by the ExtendedFSA are equally contributors to this.

A baseline of the most frequent pattern scores low. The precision never exceeds 40% and the recall is 18.65%. It is most likely that these low figures are due to the fact that the CPA corpus is not a random part of BNC; on a totally random corpus, the baseline is expected to perform better. A SVM approach which considers the right and the left context relatively to the target verb (Giuliano et all. 2009) did not

	method	*F1*
	BasicFSA	53.61
[cross validation]	ExtendedFSA	71.93
	SVM	68.58
	MostFrequent	48.12

	method	*F1*
	BasicFSA+10%	26.58
	BasicFSA+30%	37.45
[12 verbs]	ExtendedFSA+10%	45.08
	ExtendedFSA+30%	60.52
	SVM+30%	55.71
	MostFrequent	21.85

Table 8: Cross Validation and 12 Verb F1 results

perform better either. It reached an average precision of 65%, and a recall a little lower than 48%. The SVM approach works best with contexts that are bigger than the sentence, which were not available in this experiment. However the SVM figures reported above refer only to verb sense and not to pattern recognition. In Table 8 the F1 formula averaged for all verbs is presented for the 4-fold cross validation and for the set of the chosen 12 verbs respectively. A last experiment we conducted was to see how much the learned FSA matches against the raw text. The test sentences werent parsed anymore but all the nouns were considered together with their SUMO features and were sent into input to the FSA. For the 12 chosen verbs we obtained the results reported in Table 9. Using the FSA recognizer in this way means to have a deep semantic parser which provides in the same time the syntax, the dependency relationships, the senses of the words and ontological links. These are not separate operations carried in cascade, but the results of ”understanding” a verbal phrase according to the grammar associated with the respective verb. The experiments on raw text show that it is possible to develop a technique which does not necessarily make use of a parser. However, the interaction between two CCRs which are recognized in the same sentence must be first resolved in order to adopt such technique.

verb	*subject F1*	*object F1*	*verb*	*subject F1*	*object F1*
abandon	.55	.59	fence	.22	.31
accompany	.42	.34	furnish	.44	.59
acknowledge	.39	.22	launch	.58	.48
acquire	.51	.58	maintain	.39	.37
arrive	.6	.54	saddle	.34	.41
execute	.46	.61	yield	.52	.49

Table 9: Applying FSA to raw text

6 Conclusion and Further Research

The CPA is a resource that creates links between word senses and word usage. A mutual sense dependency relationship acts between the slots of a pattern. We presented a methodology for pattern learning and recognition using finite state automata. A FSA is built for each verb by using dependency chains with SUMO attribute features. In the process of learning only the relevant SUMO features are retained. The results suggest that the methodology is stable and works properly when the slots of the patterns are filled. The method is very precise for frequent senses as well as for less frequent senses. However, in order to improve the coverage, a module which handles the pronouns and proper names should be implemented. This represents the next goal for us.

The experiments we carried out suggest that the quantity of data required for training is small. We start experimenting with a training set which is built iteratively by letting the algorithm decide what is the next training example expected to help in learning the patterns. In the same vein as the original Angluins Algorithm, the learning of patterns can be carried completely automatically. The states of the obtained FSAs, although nameless, may correspond to a set of semantic types.

An important direction of work is to improve the technique of using the FSA with raw text, and shortcut the role of the parser in the architecture pipe. Our initial experiments suggest that this could be done by bootstrapping. The results obtained so far are very good and they compare positively with the ones obtained by the state of the art approaches.

7 References

D. Angluin.1987. Learning regular sets from queries and counterexamples. Inf. Comp., 75(2):87106

Briscoe, E., J. Carroll and R. Watson. 2006 The Second Release of the RASP System. In Proceedings of the COLING/ACL 2006

S. Chan, D. Roth. 2011 Exploiting Syntactico-Semantic Structures for Relation Extraction. In Proceedings of ACL 2011, Portland

M. Collins, J. Brooks. 1995. Prepositional phrase attachment through a backed-off model. In Proceedings of the Third Workshop on Very Large Corpora, pages 27–38, Cambridge.

M. Collins. 1999. Head-Driven Statistical Models for Natural Language Parsing. Ph.D. thesis, University of Pennsylvania.

B. Dorr, D. Jones. 1999. Acquisition of Semantic Lexicons in Breadth and Depth of Semantic Lexicons. Edited by Evelyne Viegas. Kluwer Press.

C. Fillmore, C. Baker, S. Hiroaki. 2002. Seeing Arguments through Transparent Structures. Proceedings of the Third International Conference on Language Resources and Evaluation (LREC). Las Palmas. 787-91

T. Dang, K. Kipper, K. Palmer, J. Rosenzweig. 1998. Investigating regular sense extensions based on intersective Levin classes. Coling-ACL98 , Montreal CA, August 11-17

C. Giuliano, A. Gliozzo and C. Strapparava .2009. Kernel Methods for Minimally SupervisedWSD. Computational Linguistics, 35:4

P. Hanks, Pustejovsky 2005. A Pattern Dictionary for Natural Language Processing, Revue Francaise de Language Appliquee, 10:2

P. Hanks.2005.Immediate Context Analysis: distinguishing meaning by studying usage. Words in Context A tribute to John Sinclair on his Retirement.

P. Hanks.2009. The Linguistics Double Helix:Norm and Exploitations. Slavonic Natural Language Processing, Brno, Masaryk University, 63-80

D. Hindle,1990. Noun classification from predicate argument structures. Proceedings of the Annual Meeting of the Association for Computational Linguistics, pp 268–275.

D. Klein C. Manning. 2003. Accurate Unlexicalized Parsing. ACL 423-430

A. Korhonen. 2002. Subcategorization Acquisition. PhD thesis published as Techical Report UCAM-CL-TR-530. Computer Laboratory

M. Lapata, C. Brew. 2004. Verb Class Disambiguation Using Informative Priors. Computational Linguistics 30:1, 45-73.

C. Leacock, G. Towell, and E Voorhes. 1993. Towards Building Contextual Representations of Word Senses Using Statistical Models. In Proceedings, SIGLEX workshop: Acquisition of Lexical Knowledge from Text, ACL.

M. Marneffe, B. McCartney, C. Manning. 2006. Generating Typed Dependency from Phrase Structure Parses. LREC 2006.

Y. Lee, H. Ng. 2002. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. Proceedings of EMNLP02, pap 4148, Philadelphia, PA, USA.

D. Li, N. Abe. 1998. Word Clustering and Disambiguation Based on Co-occurrence Data. COLING-ACL : 749-755.

D. Lin, P. Pantel. 2001. Discovery of Inference Rules for Question Answering. Natural Language Engineering 7(4): 343-360.

D. McCarthy, J. Carroll, and J. Preiss. 2001 Disambiguating noun and verb senses using automatically acquired selectional preferences. Proceedings of the SENSEVAL-2 Workshop at ACL/EACL'01, Toulouse, France.

D. McClosky, M. Surdeanu, C. Manning 2011, Event Extractions as Dependency Parsing Exploiting Syntactico-Semantic Structures for Relation Extraction. In Proceedings of ACL 2011, Portland I. Niles, A. Pease. 2001. Towards a Standard Upper Ontology. FOIS 2001

E. Pianta, L. Bentivogli, C. Girardi.2002. MultiWordnet: developing an aligned multilingual database Global WordNet, 146-154

T. Pederson. 1998. Learning Probabilistic Models of Word Sense Disambiguation .Southern Methodist University (PhD Dissertation)

T. Pederson. 2005. SenseClusters: Unsupervised Clustering and Labeling of Similar Contexts. Proceedings of the Demonstration and Interactive Poster Session of the 43rd Annual Meeting of the Association for Computational Linguistics.

A. Ratnaparkhi. 1997. A Linear Observed Time Statistical Parser Based on Maximum Entropy Models.Proceedings of the Second Conference on Empirical Methods in Natural Language Processing.

O.Popescu, B. Magnini. Sense Discriminative Patterns for Word Sense Disambiguation. SCAR workshop, NODALIDA 2007

O. Popescu 2012. Building a Resource of Patterns Using Semantic Types. Proceedings of LREC, Istanbul

A. Sun, R. Grishman 2011 Semi-supervised Semantic Pattern Discovery with Guidance from Unsupervised pattern Clusters, Exploiting Syntactico-Semantic Structures for Relation Extraction. In Proceedings of ACL 2011, Portland