Automatic Extraction of Subcategorization Frames from the Bulgarian Tree Bank

Svetoslav Marinov & Cecilia Hemming
GSLT
&
Institutionen för Språk
Högskolan i Skövde
mars@isp.his.se & cecilia@hemming.se

1st January, 2004

1 Introduction

(1) a. Teodora opened the door.
b. *Arto looked the door.

In (1-a) the verb *open* takes as an obligatory argument an NP and therefore differs from *look* in (1-b), which is an ill-formed sentence, because *look* requires a PP. These propensities of verbs to take particular kinds (and number) of arguments, is termed as *Subcategorization*.

> A *Subcategorization frame* [(SF)] is a statement of what types of syntactic arguments a verb [...] takes, such as objects, infinitives, *that*-clauses, participial clauses, and subcategorized prepositional phrases.
>
> Manning (1993:1)

SFs have been mentioned for the first time by Chomsky (1965) and in general are not restricted to verbs. As Brent (1994:434) puts it: "[...] a word may have several subcategorization frames, just as it may have several syntactic categories.".

In many areas of Computation linguistics, e.g. parsing, word sense disambiguation, etc., it is important to have a precise knowledge about the SFs of a given verb. There have been a number of attempts to extract SFs of verbs and build lexicons, each of which documenting a lesser or greater degree of success.

> Multidimensional in nature, verb subcategorization is one of the most complex type of information that a computational lexicon should provide. However, it is arguably also one of the most important type of information.
>
> Korhonen (2002:19)

In this paper we will present some of the previous work done on automatic extraction of SFs and present our own model of a system that will learn to acquire such information from parsed corpora. In our case we have looked at data from the Bulgarian TreeBank project.

2 Background

We are far from being exhaustive in this presentation of the early work dealing with automatic extraction of SFs. We only aim at giving a simple overview of the main tendencies in this area and why such information should preferably be extracted automatically and not composed manually.

Manning (1993) and Briscoe and Carroll (1997), among others, discuss the necessities of *automatically* acquiring knowledge about verbs' SFs rather than *manually* creating such lexicons. The authors agree on several issues, such as: 1) Manually compiled verb lexicons[1] are time-consuming, demand a lot of human resources and often contain mistakes; 2) Such lexicons do not exist for all languages, e.g. Bulgarian; 3) They seldom represent the up-to-date language use; and 4) They are difficult to update and 5) They are not coherent. Automatically created lexicons have, according to the authors, the benefit of 1) being based on real data, i.e. text corpora; 2) having the possibility of updating quickly; etc.

2.1 The use of raw data

Brent (1991) is one of the earliest references to acquisition of verbs' SFs. His program works on untagged text from the Wall Street Journal from which verbs are extracted first. In order to do this he uses simplistic, yet unambiguous, cues to detect the verbs. These are namely the positions of pronouns and proper names, i.e. the so called "case-marked" elements, which predict that a verb will be in the closest vicinity. After that the SFs are detected and statistically verified.

The problem with Brent's approach is that it is restricted, only 5-6 SFs are learned. Although, these are learned with a high precision, the method, relying on the simple "Case Filter"[2] cues, leaves out a lot of possibly relevant information. While Brent (1991) wants to avoid noise in the data, this can be remedied by statistical techniques (Manning 1993).

To sum up, Brent's best result shows the discovery of 40% of the verbs taking direct object, but this with only 1.5% error rate and the worst result gives 3% of all the verbs taking infinitive, yet with 3.0% error rate.

Manning (1993) criticizes this approach in terms of waste of valuable data. He also starts with raw data but first tags it using Julian Kupiec's stochastic part-of-speech tagger and then the output is sent to a finite state parser in order to extract the SFs. His system, in comparison to Brent's, distinguishes among 19 different SFs, yet still uses a variant of Brent's binomial filtering process to filter the much higher amount of false cues (i.e. initially discovered SFs). The program acquired 4900 frames for 3104 verbs from 4.1 million word corpus. Precision and recall were measured, using Oxford Advanced Learner's Dictionary as a benchmark. Thus the precision estimate for 40 verbs was 90% and the recall 43%.

Briscoe and Carroll (1997) presented a complete system that distinguishes among slightly more than 160 different subcategorization classes. The main distinctions with respect to Manning's approach concern, according to the authors, the use of global instead of strictly local syntactic information and the use of a more complex (linguistically guided) filter on the extracted patterns. The system consists of *a tagger*, *a lemmatizer*, *a parser*, *a pattern-set extractor*, *a pattern classifier*, and finally *a pattern-set evaluator* that uses statistical techniques to filter out pattern-sets for each predicate and also constructs putative lexical entries. The system's output was evaluated by comparing the results for randomly selected verbs to both manual analyses of the verbs in question and to their respective entries in dictionaries containing argument structure information[3]. The results obtained comparing the system's output to the manually made analyses gave a precision of 76,6% and a recall of 43,4%. The comparison to the dictionary entries yielded the less encouraging results of 65,7% and 35,5%, probably because the manual analyses were more suited to the test set. One weak link of the system that Briscoe and Carroll themselves point out is the filtering of patterns extracted from the data, especially for low frequency SFs. The filtering process is built on the binomial hypothesis testing originally introduced by Brent.

[1]assume lexicons of verbs with their respective SFs
[2]i.e. leave out all NPs that do not have morphological case
[3]The ANLT and the COMLEX dictionaries, see (Briscoe and Carroll 1997)

2.2 The use of parsed corpora

Sarkar and Zeman (2000) take a slightly different approach to extracting SFs. First, they use syntactically annotated data (i.e. The Prague Dependency Tree Bank) and hence the choice of language also differs from the previous approaches, all dealing with English. Czech is a free word-order language and much closer to Bulgarian than English. So the results and techniques discussed by the authors are quite relevant for us.

Sarkar and Zeman (2000) concentrate mostly on the filtering of adjuncts from the observed frames. They use 3 different statistical techniques to learn possible SFs for certain verbs, namely, *Likelihood ratio test*, *T-scores* and *Hypothesis testing*. They check all possible subsets of observed frames in order to find the best match. Their best results were achieved using the *Hypothesis testing*. Thus precision was measured 88% and recall 74%. The other two tests gave the slightly higher recall of 77%.

To sum up, Sarkar and Zeman's approach learned 137 SFs, thus competing only with Briscoe and Carroll's slightly more than 160 SFs.

2.3 Statistics

Statistical processing of the initially discovered frames is almost inevitable. Most of the authors employ similar techniques, *Log likelihood ratio tests* (Sarkar and Zeman 2000, Gorrell 1999), *T-score* (Sarkar and Zeman 2000), *Hypothesis testing* (also described as Binomial distribution) (Sarkar and Zeman 2000, Brent 1994, Manning 1993, Briscoe and Carroll 1997), *EM algorithm* (Carroll and Rooth 1998), *Clustering algorithms* (Basili and Vindigni 1998).

Of the above methods *Hypothesis testing* has been used most widely and probably most successfully. Yet, a problem with it is that it assumes an uniform error likelihood of all verbs disregarding their rather zipfian-like distribution. It disregards the correlation between the conditional distribution of SFs given a predicate and the unconditional distribution independent of a specific predicate (Korhonen 2002). Sarkar and Zeman (2000) also mention this and propose the use of *multinomial distribution*.

Gorrell (1999) also argues against the use of binomial distribution for the sake of Log likelihood ratio test. The argumentation is that many valid SFs in English are rare and the former technique filters them out quite improperly. Yet, the results which Gorrell (1999) gets, show that the hypothesis test is after all to be preferred for the task of learning SFs. Sarkar and Zeman (2000) also prove this. However, there are other statistical techniques that can be employed (Dunning 1993).

3 BulTreeBank

For the present purpose we chose to look at data from the Bulgarian Tree Bank project (Simov et al. 2002a). We were offered[4] a preliminary version of the parsed corpus. The file consists of 580 sentences, fully parsed in HPSG formalism and each word carries a rich part of speech tag. The original xml file was transliterated for the sake of ease in processing and viewing under different operating systems. An exemplary sentence is given below[5]:

(2) Ne mi e do smyah.
Not me-refl is to laughter
"I'm not in the mood to laugh."

```
<s index="d001.s1"><class></class><text>Ne mi e do smyah.
</text><analysis><S><Discourse><InDiscourse>
</InDiscourse><OutDiscourse></OutDiscourse></Discourse>
<CoIndex></CoIndex><VPC><V><T><w><ph>Ne</ph>
```

[4]Many thanks to Kiril Simov for providing us with the necessary xml files, dtds and relevant descriptions
[5]new lines added here for the sake of visibility

```
<aa>T</aa><ta>T</ta></w></T><V><Pron><w><ph>mi</ph>
<aa>Ncnsi;Pp-d1s-t;Pso-1--t;T;Vpit+f-o2s;Vpit+f-o3s</aa>
<ta>Pp-d1s-t</ta></w></Pron><V><w><ph>e</ph>
<aa>I;T;Vx---f-r3s</aa><ta>Vx---f-r3s</ta></w></V>
</V></V><PP><Prep><w><ph>do</ph>
<aa>Ncnsi;R</aa><ta>R</ta></w></Prep><N><w>
<ph>smyah</ph><aa>Ncmsi;Vpii+f-o1s</aa><ta>Ncmsi</ta>
</w></N></PP></VPC><pt>.</pt></S>
</analysis><source></source></s>
```


We were recommended to work with the special tool for corpora development - CLaRK system (Simov et al. 2002b). However, since CLaRK did not yet have the possibility to extract the necessary information about verbs, as well as the time consuming process to set ourselves into the software, we decided to find other ways to preprocess the data in order to extract the SFs of some of the verbs in the corpus.

Of the rich information in the above sentence, we considered the following excerpt to be the most important and relevant for the present task, thus reducing the above sentence to:

```
<S <VPC <V <T Ne <ta T </T <V <Pron mi <ta Pp-d1s-t </Pron <V e <ta
Vx---f-r3s </V </V </V <PP <Prep do <ta R </Prep <N smyah <ta Ncmsi
</N </PP </VPC . </S
```

In addition we have preserved information about coindexation, e.g. in cases of topicalization; discontinuous constituents, pro-drop, etc. see examples (4) and (5) below. We work with approximately 300K data.

4 Two experiments

We decided to conduct two experiments with the available data. One using a POS-tagged corpus and the other one using a fully parsed input.

For the first task, using perl scripts we filtered the original XML-file, leaving only one word per line with its respective POS-tag, i.e. reducing it to the standard input to Cass, Abney's partial parser (Abney 1996), see example sentence below.

```
<s>
Ne      T
mi      Pp-d1s-t
e       Vx---f-r3s
do      R
smyah   Ncmsi
.       .
</s>
[I am not in the mood to laugh.]
```

The morphologically rich tags were then substituted by the Penn Treebank tags and this constituted the input to the parser. Certainly, the substitution of tags can be criticized, since a lot of valuable information was lost. An alternative with which we could work in the future is to write a Cass-style grammar to interpret the Bulgarian tags. Thus we will achieve better parsing result, in comparison to the present, very noisy parsed data.

Since Cass offers the possibility to extract the arguments of verbs, we used it to get a verb-argument list, a little example is given below:

```
e       :subj mi  (is, subj:me)
*do     :subj %name (wrongly parsed; by, subj: name)
padnala :na %name (fall, subj:proper name)
dimeshe           (was smoking)
govori  :obj tolkova    :subj Tya  (talk, obj:so much, subj:pronoun)
tryabvashe      :subj kolkoto      (was necessary, subj:so much)
```

We intend to filter this output using the standard log likelihood ratio test and compare the results to the results we get from the second experiment with the fully parsed data.

5 Outline of the system for extracting SFs from the BulTreeBank

Here we outline the implementation of a system for learning SFs for Bulgarian verbs from the BulTreeBank, the Bulgarian HPSG-parsed corpus (Simov et al. 2002a).

We have implemented a module, in the Oz programming language (van Roy and Haridi to appear: March 2004), that extracts all verbs with all their possible cues from the corpus data. Because of the naïvety of the algorithm, this is a very noisy data. The SFs often contain more arguments than necessary, e.g. see (3):

(3)

```
obikalyashe: N: 1 times (walk around)
kazhi: Pron CoordP CLDA VPC Pron PP N: 1 times (say)
v'rvyah: Pron Pron: 1 times (walk)
uvelichat: NPA N PP NPA NPA N PP N PP NPA Pron N: 1 times (enlarge)
vklyuchva: NPA N PP NPA Pron N: 1 times (include)
s'biraha: PP NPA NPC N N NPA: 1 times (gather)
umira: N: 1 times (die)
```

Similarly to (Sarkar and Zeman 2000), we collect both arguments and adjuncts. We have also collected the external argument, i.e. the subject of a VP-clause. Since Bulgarian is a relatively free word order language the subject of a VP-clause can appear either somewhere to the right or to the left of the main verb. This is also true about the direct and indirect objects.

(4) Rabotata im, kaži-reči, e cjalata naša.
Work-def theirs, almost, is whole-def ours
"All their work is almost ours."

(5) Ot Sofia sreštnah m'ža.
From Sofia I met man-def
"I met the man from Sofia."

The next step in our implementation is to find the lemma for each verb form that appears in the collection. This we did by hand but the necessary module is under construction. After the lemmatization process, we remain with data for verb-lemmas in stead of individual verb forms.

Unlike (Sarkar and Zeman 2000), however, we decided to experiment with the *Binomial Log Likelihood Ratio* test as a filter for implausible frames. As our SF-extractor returns too many possibilities as arguments in a frame, all the possible frames were shortened to only 6 elements, if longer than this number. The Log Likelihood statistics were then calculated on this input. Some results are shown in Figure 1.

The results in Fig. 1 are not very promising. This is dependent on 2 factors - 1) the too general filters of the SF-extractor and 2) the sparse data. We would like to take the experiment with Log Likelihood Ratio test a step further and test with subsets of frames, á la Sarkar and Zeman (2000). We will face the same problem as they, namely the lack of electronically available valency lexicons. Therefore, the evaluation has to be done manually. We cannot rely on using additional, unseen data, as there is no such, yet this is what

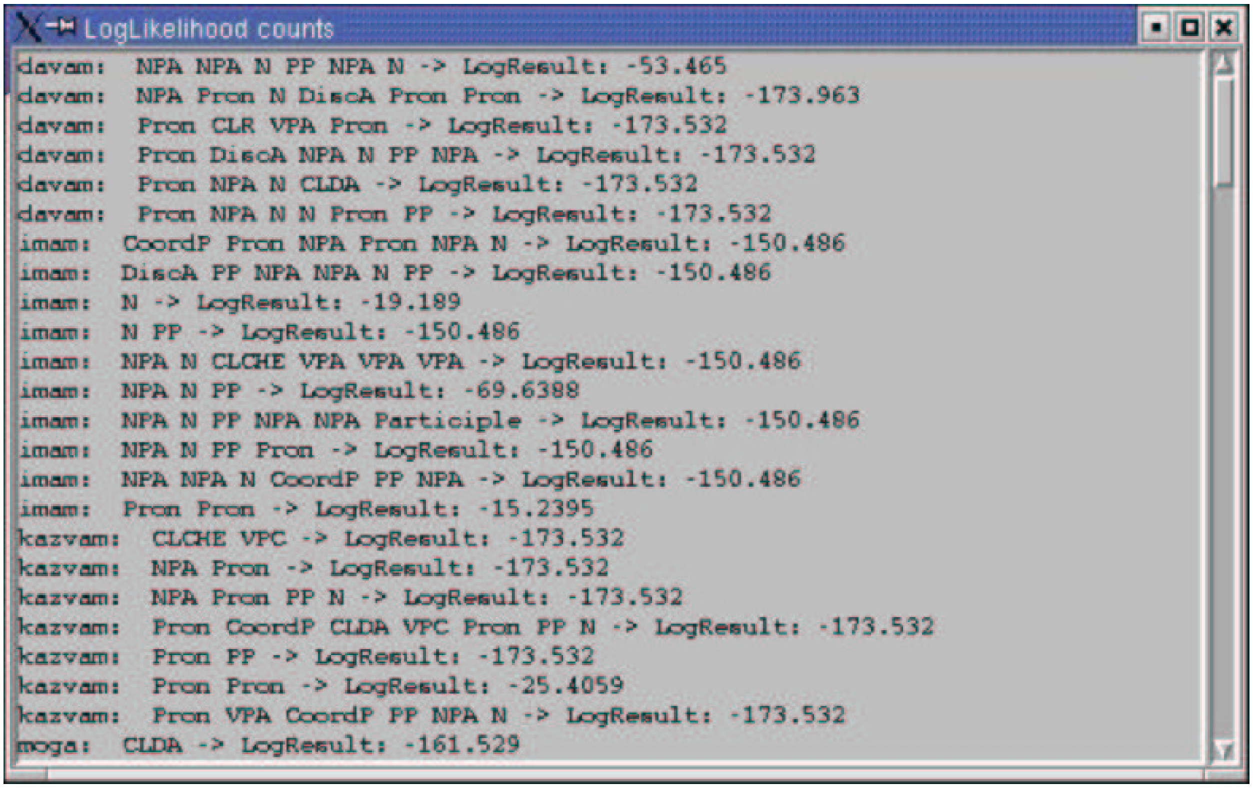

Figure 1: Log Likelihood Ration calculations

the evaluators of the Czech results have used.

There is another difference with respect to the former work, since the authors use a treebank annotated with dependency relations. Our data, however, are annotated with HPSG-style relations. This gives us clues for possible SFs, e.g. the following part of a sentence, especially the "<VPC"-tag will give us the information that we have a verb ("<V e" = is) that takes a complement ("<N smyah = laughter). A dependency annotated corpus do *not* give out such subcategorization information.

```
<VPC <V <T Ne <ta T </T <V <Pron mi <ta Pp-d1s-t </Pron <V e <ta
Vx---f-r3s </V </V </V <PP <Prep do <ta R </Prep <N smyah <ta Ncmsi
</N </PP </VPC .
```

We would like to use this information, if possible. We have devised an algorithm which will be implemented in order to test whether it would give us better predictions for possible SFs.

```
Main:

For each sentence in the corpus
  For each verb
    If it exists
      Compare the new SF with the existing ones
        If it exists
          Add 1 to its count
        Else
          Add a new SF to the list of previous frames
```

```
9.           Initialize the count of this frame to 1
10.        End
11.     Else
12.       Find the verb's SF
13.       Add it to the repository (a dictionary)
14.     End
15.   End
16. End

Subroutine:

Find_SF:

1. Find the verbs internal arguments
2. Make a verb list with the verb being a head and its internal
     arguments (in order of appearance) the tail
3. Find its external argument
4. Add it to the verb list
5. End
```

6 Conclusion

In this paper we presented the major works in the area of automatic learning of verbs' SFs. Most of these used raw data which was further processed in order to capture the relevant information. Almost all works were concerned with English, except Sarkar and Zeman (2000) and Basili and Vindigni (1998), dealing with Czech and Italian respectively. Similar statistical methods were used in most of these works, namely the Hypothesis testing. Despite its obvious drawbacks, this method turns out to be most successful in filtering out irrelevant frames.

We have attempted two tests on a Bulgarian parsed corpus and have shown preliminary results of finding SFs using Abney's partial parser on tagged data and our own SF-extractor on parsed data. Still some work remains to be done with respect to better extraction, filtering and learning frames from the parsed data. It is also worth looking at what could further be acquired from the morphologically rich tags of the BulTreeBank. It would be interesting to see how one can discover relationships between a verb's different SFs, and classify them according to their possible SF sets.

References

Abney, Steven. 1996. Partial parsing via finite-state cascades. *Journal of Natural Language Engineering* 2(4):337–344.

Basili, Roberto, and Michele Vindigni. 1998. Adapting a subcategorization lexicon to a domain. In *Proceedings of the ECML'98 Workshop* TANLPS: Towards adaptive NLP-driven systems: linguistic information, learning methods and applications. Chemnitz, Germany.

Brent, Michael R. 1991. Automatic acquisition of subcategorization frames from untagged text. In *Meeting of the Association for Computational Linguistics*, 209–214. URL `citeseer.nj.nec.com/brent91automatic.html`.

Brent, Michael R. 1994. Surface cues and robust inference as a basis for the early acquisition of subcategorization frames. *Lingua* 92:433–470.

Briscoe, Ted, and John Carroll. 1997. Automatic extraction of subcategorization from corpora. In *Proceedings of the 5th ACL Conference on Applied Natural Language Processing, Washington, DC.*. URL `citeseer.nj.nec.com/ted97automatic.html`.

Carroll, Glenn, and Mats Rooth. 1998. Valence induction with a head-lexicalized PCFG. In *Proceedings of the 3rd Conference on Empirical Methods in Natural Language Processing (EMNLP 3)*. Granada, Spain.

Chomsky, Noam. 1965. *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.

Dunning, Ted. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics* 19(1):61–74.

Gorrell, Genevieve. 1999. Acquiring subcategorization from textual corpora. Master's thesis, Cambridge University.

Korhonen, Anna. 2002. Subcategorization acquisition. Technical report, University of Cambridge, Computer Laboratory. URL `citeseer.nj.nec.com/article/korhonen02subcategorization.html`.

Manning, Christofer. 1993. Automatic acquisition of a large subcategorization dictionary from corpora. In *Proceedings of the 31st ACL*, 235–242. URL `http://www-nlp.stanford.edu/~manning/papers/subcats.ps (as of November 2003)`.

van Roy, Peter, and Seif Haridi. to appear: March 2004. *Concepts, techniques and models of computer programming*. MIT Press.

Sarkar, Anoop, and Daniel Zeman. 2000. Automatic extraction of subcategorization frames for czech. In *Proceedings of COLING 2000*. Saarbrücken, Germany. URL `http://www.sfu.ca/~anoop/papers/pdf/coling00_final.pdf (as of November 2003)`.

Simov, Kiril, Gergana Popova, and Petya Osenova. 2002a. HPSG-based syntactic treebank of Bulgarian (BulTreeBank). In *A rainbow of corpora: Corpus linguistics and the languages of the world*, ed. Andrew Wilson, Paul Rayson, and Tony McEnery, 135–142. Lincon-Europa, Munich.

Simov, Kiril, Alexander Simov, Milen Kouylekov, and Krassimira Ivanova. 2002b. CLaRK - an XML-based system for corpora development. Technical report, Linguistic Modelling Laboratory - CLPPI, Bulgarian Academy of Sciences.