Natural Language Engineering **9** (1): 39–48. © 2003 Cambridge University Press
DOI: 10.1017/S1351324903003097 Printed in the United Kingdom

Application of finite-state transducers to the acquisition of verb subcategorization information

I. ALDEZABAL, M. ARANZABE, K. GOJENOLA,
M. ORONOZ, K. SARASOLA

*IXA group, Department of Computer Languages and Systems, University of the Basque Country,
649 P.K., 20080-Donostia, Spain*
e-mail: {jibalroi, jibarurm, jipgogak, jiboranm, jipsagak}@si.ehu.es

A. ATUTXA

Department of Linguistics, University of Maryland, College Park, MD 20742, USA
e-mail: sener@wam.umd.edu

(*Received 27 March 2002; revised 31 October 2002*)

Abstract

This paper presents the design and implementation of a finite-state syntactic grammar of Basque that has been used with the objective of extracting information about verb subcategorization instances from newspaper texts. After a partial parser has built basic syntactic units such as noun phrases, prepositional phrases, and sentential complements, a finite-state parser performs syntactic disambiguation, determination of clause boundaries and filtering of the results, in order to obtain a verb occurrence together with its associated syntactic components, either complements or adjuncts. The set of occurrences for each verb is then filtered by statistical measures that distinguish arguments from adjuncts.

1 Introduction

This paper shows the application of a finite-state syntactic grammar to the acquisition of verb subcategorization information from a corpus of newspaper texts. In the absence of a full parser for Basque, as it happens in many languages, we recurred to shallow parsing, as in Abney (1997). This allowed us to create a parser with a limited effort at the cost of losing coverage. A first parsing module groups words that form adjacent syntactic units, such as NPs, PPs, and sentential complements. The output of this phase is not directly usable, due mainly to ambiguities and to the difficulty of linking dependents with verbs. For that reason, a finite-state parser performs syntactic disambiguation, determination of clause boundaries and filtering of the results. The system has been applied to Basque, which has as its main characteristics being agglutinative and having basically constituent-free order. The result of the parsing phase is a set of verb instances together with their syntactic dependents (arguments or adjuncts) for each sentence in the corpus. The information thus obtained was the input to two different statistical filters which were

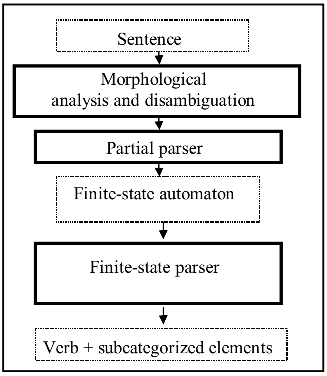

Sentence
Morphological analysis and disambiguation
Partial parser
Finite-state automaton
Finite-state parser
Verb + subcategorized elements

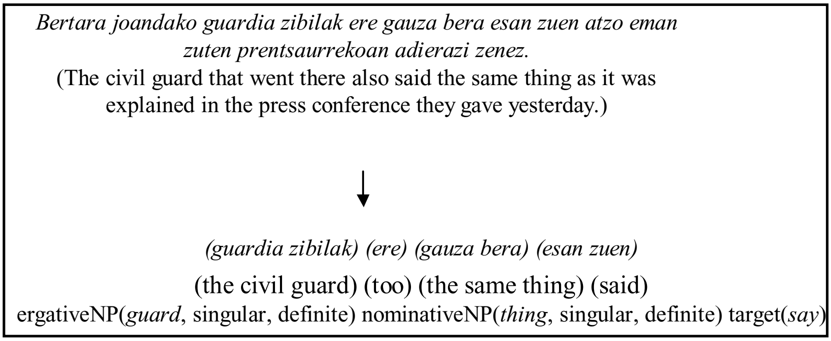
Fig. 1. Overview of the system.

tested on their adequacy to distinguish arguments from adjuncts, with the aim of automatically obtaining verb subcategorization frames.

The rest of the paper is organized as follows. Section 2 shows the initial parsing system we have used, detailing its main components. Section 3 examines the finite-state grammar applied to the extraction of subcategorization information. Section 4 presents some evaluation results. Finally, Section 5 presents related work on lexical acquisition in the form of verb subcategorization information.

2 System architecture

Figure 1 shows the architecture of the system. First, it performs morphological analysis based on two-level morphology (Alegria, Artola, Sarasola and Urkia 1996) and disambiguation using the Constraint Grammar (CG) formalism (Karlsson, Voutilainen, Heikkila and Anttila 1995; Ezeiza, Alegria, Arriola, Urizar and Aduriz 1998). These two steps are fundamental for the processing of agglutinative languages. As a second step, a partial parser is applied (Abney 1997), which recognizes basic syntactic units (NPs, PPs and several types of subordinate sentences). Currently we can employ two different partial parsers, one of them using a unification grammar (Aldezabal, Gojenola and Sarasola 2000) and the other one based on the CG formalism (Ezeiza, Alegria, Arriola, Urizar and Aduriz 1998). Although based on very different formalisms, both obtain similar results, in terms of coverage and ambiguity. However, there is a long way when moving from basic syntactic units to full parsing. Many ambiguities, which can be manageable locally, give thousands of combinations at the sentence level (see Figure 3), as a result of both morphological ambiguity (1.19 interpretations per word-form after morphological disambiguation), syntactic ambiguities introduced by the partial parser, and attachment ambiguities between verbs and their dependents. This problem has been partially solved for several languages where full sentence parsers have been enriched with statistical information or by adding subcategorization information (Briscoe and Carroll 1997; Korhonen 2001). In languages like Basque, neither annotated treebanks nor information on verb subcategorization are available. As any of these elements would



Bertara joandako guardia zibilak ere gauza bera esan zuen atzo eman zuten prentsaurrekoan adierazi zenez.
(The civil guard that went there also said the same thing as it was explained in the press conference they gave yesterday.)

↓

(guardia zibilak) (ere) (gauza bera) (esan zuen)
(the civil guard) (too) (the same thing) (said)
ergativeNP(*guard*, singular, definite) nominativeNP(*thing*, singular, definite) target(*say*)

Fig. 2. Example of an input sentence and its corresponding output.

be very expensive, we took the option of developing a finite-state parsing module which performs syntactic disambiguation and filtering of the results, with the aim of obtaining subcategorization information that will be used in subsequent works to enrich the lexical database.

3 A finite-state grammar for the acquisition of verb subcategorization instances

These are the main operations performed by the finite-state parser:

- Clause recognition. To extract verb subcategorization information, a set of rules examine the context of the target verb and define the scope of the clause to which the disambiguation operations will be applied.
- Disambiguation. Morphological and syntactic ambiguities are the cause of obtaining multiple readings per sentence. This disambiguation process is similar to that of CG disambiguation with the advantage of being able to reference syntactic units wider than the word, which must be defined in a roundabout manner in the word-based CG formalism. A special kind of disambiguation specific to the task of extracting verb subcategorization information is the filtering of several non-interesting syntactic tags. For example, the noun/adjective ambiguity can be ignored when acquiring verb subcategorization information, as we are interested in the syntactic category and the grammatical case, the same in both alternatives.

Figure 2 shows the application to a sentence, in the context of analyzing the verb *esan* (to say). The result has been simplified, because both the input and output are presented as text, rather than as automata containing feature-value pairs that represent syntactic components, as in Figure 3 (actually each unit also contains its corresponding morphosyntactic tags, such as case, number, or subordination type). Taking a sentence as input, the clause corresponding to the target verb is delimited first. Then, after several disambiguation steps, the syntactic units that appear with the verb are selected, each with its associated syntactic features.

To implement the finite-state grammar we have applied operations on regular expressions and relations, among them composition and replacement, using the Xerox

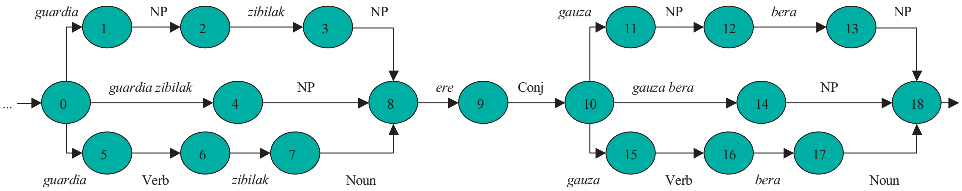


Fig. 3. Automaton corresponding to a part of the sentence in figure 2: "… *guardia zibilak ere gauza bera …*".

Finite State Tool (XFST, (Karttunen, Chanod, Grefenstette and Schiller 1996)). We use both ordinary composition and the lenient composition operator (Karttunen 1998). This operator allows the application of different eliminating constraints to a sentence, always with the certainty that when some constraint eliminates all the interpretations, then the constraint is not applied at all, that is, the interpretations are *rescued*, making the system robust. The operator was first proposed to formalize Optimality Theory constraints in phonology. As Karttunen points out, it also provides a flexible way to enforce linguistic or empirical constraints in syntactic disambiguation. Gerdemann and van Noord (2000) have demonstrated how a matching approach to the same problem, without using lenient composition, improves the results, as it eliminates the need for an upper limit on the counting, while at the same time minimizes the resulting transducer. However, as lenient composition was already implemented as part of the XFST set of operators, we did not test this latter option. In the next subsections we will describe the different parts of the finite-state grammar.

3.1 Clause recognition

To extract verb subcategorization information, a group of rules will examine the context of the target verb and define the scope of its associated clause. The global ambiguity is considerably reduced if only the clause corresponding to the target verb is considered. In the following steps the disambiguation operations will be applied to this clause.

For example, the following rule will insert a clause boundary to the left of the target verb just before one or more intermediate NPs or PPs if there is a subordinate modifier to their left, which in Basque cannot be part of the clause corresponding to the target verb[1]:

```
define MarkLeftSubordinate [[NP | PP | NegativeParticle]* TargetVerb]
                                  @-> ‘‘{{’’ ... || SubordinateModifier _ ;
```

When applied to the sentence of Figure 2, the delimiter "{{" will be inserted to mark the beginning of a clause ("*Bertara joandako* {{ *guardia zibilak ere gauza bera esan* …"), because the verb *joandako* contains a subordinate modifier that is out of the scope of the target verb. Twenty-two rules have been defined for the task of

[1] A rule of the form A @− > B … C || D _ E will insert the tags B and C to either side of A in the context D A E.

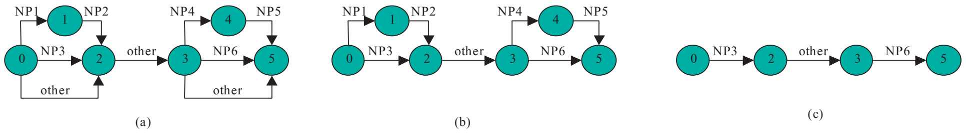


Fig. 4. Application of disambiguation constraints. (a) Initial automaton, (b) remaining alternatives after selecting the maximal projections, (c) remaining alternative after selecting the minimal number of NPs.

delimiting boundaries, which are sufficient to obtain high precision. The compilation of the previous rule produces a transducer with 482 states and 12,221 arcs. As most of the rules are as complex as the one presented, they cannot be composed into a single automaton, because of its prohibitive size. For that reason, the clause filters are applied sequentially.

3.2 Disambiguation: longest match selection

A usual kind of ambiguity is produced when there are alternative readings of a sentence where some of the syntactic units can be either independent or can also be included inside bigger units. Figure 4(a) is a simplification of the automaton in Figure 3, which shows an example where there are several arcs giving NPs[2], while some others cannot be analyzed (they have been simplified in the picture, marked as "other"). This kind of unanalyzed element is relatively frequent and corresponds to punctuation marks, unknown words or portions not covered by the partial grammar. After examining a number of sentences, we decided to apply two heuristics:

- Take the readings with NPs covering most of the sentence (maximal projections). Figure 4(b) shows how after applying this heuristic the number of readings is reduced from nine to four. Although at first glance it could seem that eliminating any "other" interpretation could suffice, this would not work, because it would eliminate all the readings, as there is one "other" path (from state 2 to 3) that must be followed by all readings. For the selection of maximal projections, we applied the following constraints sequentially:

  — Take the reading(s) that contains no "other" arcs.
  — Take the reading(s) containing at most one "other" arc.
  — . . .
  — Take the reading(s) containing at most N "other" arcs[3].

  As the constraint composition with any sentence results in a null output because no reading can meet all of them, the constraints will be applied

[2] Although we will illustrate the problem using NPs, the method has been equally applied to other kinds of syntactic units, such as verb chains, PPs or sentential complements. In fact, we treat both NPs and PPs as NPs, because in Basque they have the same syntactic structure, differing only in syntactic case.

[3] The value N must be a reasonable constant. With long sentences, $N = 20$ is enough.

sequentially using lenient composition (.O. in the XFST tool), so that the reading(s) with the minimum number of "other" arcs will be selected[4]:

```
define Sentence0 Sentence  .O. ~$Other
define Sentence1 Sentence0 .O. ~[[$Other]^>1]
...
define SentenceN SentenceN-1 .O. ~[[$Other]^>N]
```

Figure 4(b) shows the result after applying these constraints to the automaton in figure 4(a).

- Take the longest NPs (minimal number of maximal projections). The number of maximal projections is minimized to obtain a preference for more extended linguistic analyses over a series of smaller ones. To select the readings with the longest NPs we apply a similar approach, selecting the readings with the minimum number of NPs. Again, the lenient composition operator must be used. Figure 4(b) shows four readings, but the alternative containing only NP3 and NP6 will be usually preferred (Figure 4(c)).

The heuristics have been applied in a reductionistic manner, discarding the readings that do not meet the required constraints. The two sets of restrictions must be applied sequentially, because otherwise the results would not be the intended ones. For example, if we first applied the second heuristic to the automaton given in Figure 4(a), it would incorrectly take two readings (one of them would be composed by NP3 plus two "other" arcs) when the desired result should contain at least two NPs. The heuristics for the selection of the longest match perform satisfactorily (see section 4 for a preliminary evaluation). In a first comparison to English, one could ask if the heuristics would imply that some patterns, such as V NP PP, would always be reduced to V NP, missing the pattern NP PP. Due to the characteristics of Basque syntax this problem is not present, as any PP linked to an NP appears before it and uses the special genitive suffixes *-ko* and *-en*, while those PPs linked to the main verb use special case markers. Obviously, this kind of problem has been avoided in part due to the specific work of extracting subcategorization information, and will have to be dealt with when developing a full syntactic analyzer.

3.3 General disambiguation constraints

We have also developed a number of constraints that try to reduce several types of ambiguity. For example, a common case of ambiguity between NP-Subject-Ergative-Singular and NP-Object-Nominative-Plural (as in *gizonak*-the man(Subject)/the men(Object)) can be resolved using information about agreement when the verb is finite. Another ambiguous situation concerns the distinction in subordinated sentences between an indirect interrogative clause and a relative clause, which many times can be resolved due to the presence of an interrogative pronoun. These constraints must also be applied using the lenient composition operator, so that

[4] The following operators are used: ∼ (complement), $ ("contains" operator) and X ^ > Y (Y or more repetitions of X) .

no constraint will produce a null result. For example, the following rule eliminates sentence readings containing a singular NP in nominative case and a finite verb that requires a plural nominative NP, due to agreement in number:

```
define ApplyAgreement1 ~$[ VerbNomPlural ?* NPNomSingular ];
```

The constraint applies after correctly delimiting the clause, because the presence of multiple verbs would make the restriction highly inaccurate. There are sixteen constraints treating agreement, solving only part of this kind of ambiguity (for example, non-finite verbs lack agreement, hence the constraints do not apply).

4 Evaluation

The resulting finite-state grammar contains more than 300 rules. It has been applied to a corpus of 111,000 sentences from newspaper texts, totalling 1,337,445 words. When dealing with unrestricted texts there are several extra difficulties added to the problem of ambiguity, such as multiword lexical units, unknown words, proper names, spelling errors and long sentences (as each sentence contains an instance of the target verb together with other main or subordinate clauses, delimiting the exact boundaries of the clause corresponding to the target verb is a difficult task). For evaluation of the parser we took a set of previously unseen 150 sentences (Aldezabal, Gojenola and Sarasola 2000). We measured precision (# of correctly selected dependents/all the elements returned) and recall (# of correctly selected dependents/actual dependents in the sentence), applied to each instance of the target verb and its corresponding complements/adjuncts. Although there is always a balance between precision and recall, we tried to maximize precision, sometimes at the cost of lowering recall. We consider the results satisfactory, with 87% precision and 66% recall. We examined the causes of the errors manually, concluding that about half of them can be improved by simple refinements of the lexicon and the grammars (both the partial grammar and the finite-state grammar), while the rest would require qualitative changes on the grammars (incorporation of subcategorization information, etc.).

4.1 The argument selection phase

This task consists of finding for each verb correct syntactic frames, free from adjuncts. What we obtained from parsing were analyzed contexts (combinations of dependents) corresponding to verbs. To obtain subcategorization frames from these contexts we have to identify and eliminate adjuncts and errors resulting from parsing (since precision and recall were not of a 100%). A way to tackle this problem is to apply statistical filters to discriminate arguments from other elements (either adjuncts or parsing errors). We applied two statistical measures: Mutual Information (MI), and Fisher's Exact Test. These tests are broadly used to discover associations between words (Manning and Schutze 1999). Based on theoretical grounds, one would expect to find a high association between a verb and an argument and a low association between a verb and an adjunct (Aldezabal, Aranzabe, Atutxa,

Table 1. *Results of evaluation (non-contextualized)*

	Precision (%)	Recall (%)	F-score (%)
MI	62	50	55
Fisher	64	44	52

Table 2. *Results of evaluation (contextualized)*

	Precision (%)	Recall (%)	F-score (%)
MI	93	97	95
Fisher	93	93	93

Gojenola and Sarasola 2002). The subcategorization system is fine-grained in that it is not limited to syntactic categories such as NP or PP, but to 48 different types of grammatical cases (absolutive, dative, inessive, ...) and sentential complements. We evaluated the results against manually annotated data, following two approaches. For the first approach we used no context, so the annotators could not derive the sense of the verb among its different senses. We selected 10 verbs and, for each of them, we extracted from the corpus the list of all case markers of its dependents. We provided 4 human annotators with this bare list of verbs and cases, and they marked each case as either a case introducing an argument or an adjunct. So the annotators labeled the dependents considering multiple senses of the verb, even those not appearing in the corpus, since there was no context to disambiguate the specific meaning of the verb (we will call this the non-contextualized evaluation). Table 1 shows the results after comparing the values obtained for each verb-case pair by the annotators (gold standard) with the ones obtained by the system.

Table 2 shows the result when providing the annotators with no list but a set of sentences corresponding to the 10 verbs. The annotators extracted the verb-case pairs for each sentence, and using the sentence as context they could derive the particular sense of the verb. As we used different parts of the same corpus for learning and evaluating, the semantics are restricted to that of the corpus. We performed a simple evaluation, calculating precision and recall of the system over each argument marked by the annotators. This can give an estimate of the extent to which the information obtained could be useful for a parser applied to that corpus.

The origin of the differences in Tables 1 and 2 comes, on the one hand, from semantics. The former evaluation was not contextualized, while the latter used the sentence context. Many times the different senses of the same verb do not show the same arguments. Hence, if the number of senses increases the number of arguments increases too, and the task becomes more difficult. On the other hand, there are also statistical reasons. For the first experiment all possible arguments were evaluated, including the less frequent ones, whereas in the second experiment only the possible arguments found in the piece of the corpus for evaluation were used. Often, the possible arguments found were the most frequent ones. Therefore, as statistical measures performs better on frequent cases than on non frequent ones, we obtained better results on the second evaluation.

5 Related work on the acquisition of subcategorization information

Concerning the acquisition of subcategorization information, there are proposals ranging from manual examination of corpora to fully automatic approaches. Manning (1993) presents an approach starting from raw corpora. He uses a stochastic tagger and a finite-state parser to obtain instances of verbs with their dependents (either arguments or adjuncts), and then a statistical filtering phase produces subcategorization frames (from a set of previously defined 19 frames) for each verb. Manning's system learns from contexts with just one PP (maximum) per verb. Our system is more ambitious, as it learns from contexts with up to 5 PPs. Furthermore, he considers syntactic categories, while we deal with cases (more precisely, we distinguish 48 different kinds of cases), hence the combination number increases. Briscoe and Carroll (1997) describe a grammar based experiment for the extraction of subcategorization frames with their associated relative frequencies, obtaining 76.6% precision and 43.4% recall (non-contextualized evaluation). They use ANLT and COMLEX dictionaries (containing subcategorization information) to filter the contexts obtained by their parser. For our work, we could not make use of any previous information on subcategorization, because no Basque dictionary contains any, apart from the traditional transitive/intransitive distinction. Sarkar and Zeman (2000) report results for Czech, a free word order language. The input to the system is a set of manually annotated sentences, where each verb is linked with its dependents. Comparing this approach to ours, their data does not come from raw corpus, thus they do not deal with the problem of noise coming from the parsing phase. Their main limitation comes from relying on a treebank, which is an expensive resource. Regarding the parsing component, the systems presented so far are heterogeneous. While Manning uses simple parsing techniques, Briscoe and Carroll use a sophisticated full parser. Our system can be placed between the two approaches. Our shallow parsing results relie on a robust morphological analysis and disambiguation, while the finite-state grammar we used for parsing is very sophisticated compared to that used in Manning (1999), which contained 14 states and 100 transitions.

6 Conclusion

In the literature, finite-state processing has been applied to a variety of tasks, ranging from morphology to syntax. This work presents the application of a finite-state grammar to the specific task of detecting verbs and their associated sentence components. The application works at a level higher than morphology, as previously built syntactic units must be dealt with, but only with a restricted subset of syntax: the relation of verbs with their adjuncts and complements. The task is complex because many problems must be taken into account, such as ambiguity and determination of clause boundaries, among others. This way, without developing a full syntactic parser, we have been able to obtain useful results which are accurate enough. The finite-state grammar provides a modular, declarative and flexible workbench to deal with the graph of syntactic components. It establishes the application of empirical, corpus-oriented facts about morphological/syntactic ambiguity, versus the more general facts on linguistic well-formedness encoded in the previously applied partial grammar. Although the finite-state grammar was specifically designed for that concrete application, many rules express general linguistic facts and are reusable, so that we plan to include them in a more general syntactic finite-state grammar, together with the subcategorization information that is being acquired.

Acknowledgements

This research was supported by the European Commision (project MEANING IST-2001-34460), the Spanish Ministry of Science and Technology (project Hermes TIC2000-0335-C03-03), the Department of Education, Universities and Research of the Basque Country, the University of the Basque Country and the Interministerial Commision for Science and Technology (CICYT).

References

Alegria, I., Artola, X., Sarasola, K. and Urkia, M. (1996) Automatic morphological analysis of Basque. *Literary and Linguistic Computing* **11**(4).

Abney, S. P. (1997) Part-of-speech tagging and partial parsing. In: Young, S. and Bloothooft, G., editors, *Corpus-Based Methods in Language and Speech Processing*. Kluwer, Dordrecht.

Aldezabal, I., Gojenola, K. and Sarasola, K. (2000) A bootstrapping approach to parser development. *Proceedings International Workshop on Parsing Technologies*, Trento, Italy.

Aldezabal, I., Aranzabe, M., Atutxa, A., Gojenola, K. and Sarasola, K. (2002) Learning argument/adjunct distinction for Basque. *SIGLEX Workshop (ACL2002)*, Philadelphia.

Briscoe, T. and Carroll, J. (1997) Automatic extraction of subcategorization from corpora. *Proceedings Conference on Applied NLP*, Washington.

Ezeiza, N., Alegria, I., Arriola, J. M., Urizar, R. and Aduriz, I. (1998) Combining stochastic and rule-based methods for disambiguation in agglutinative languages. *Proceedings COLING-ACL*, Montreal, Canada.

Gerdemann, D. and van Noord, G. (2000) Approximation and exactness in finite-state Optimality Theory. *Proceedings Fifth Workshop of the ACL Special Interest Group in Computational Phonology (SIGPHON) (COLING 2000)*, Luxembourg.

Karlsson, F., Voutilainen, A., Heikkila, J. and Anttila, A. (1995) *Constraint Grammar: a Language-Independent System for Parsing Unrestricted Text*. Mouton de Gruyter.

Karttunen, L., Chanod, J.-P., Grefenstette, G. and Schiller, A. (1996) Regular expressions for language engineering. *Natural Lang. Eng.* **2**(4), 305–328.

Karttunen, L. (1998) The proper treatment of optimality in computational phonology. *Proceedings International Workshop on Finite State Methods in NLP*, Ankara, Turkey.

Korhonen, A. (2001) Subcategorization acquisition. PhD Thesis, University of Cambridge.

Manning, C. D. (1993) Automatic acquisition of a large subcategorization dictionary from corpora. *Proceedings 31st Conference of the ACL.*

Manning, C. D. and Schutze, H. (1999) *Foundations of Statistical NLP*. MIT Press.

Sarkar, A. and Zeman, D. (2000) Automatic extraction of subcategorization frames for Czech. *Proceedings COLING-2000*, Saarbrucken, Germany.