

Subject and Object Dependency Extraction Using Finite-State Transducers

Salah Aït-Mokhtar, Jean-Pierre Chanod
Rank Xerox Research Centre, 6 Chemin de Maupertuis
F-38240 Meylan, France
{Ait,Chanod}@grenoble.rxrc.xerox.com

Abstract

We describe and evaluate an approach for fast automatic recognition and extraction of subject and object dependency relations from large French corpora, using a sequence of finite-state transducers. The extraction is performed in two major steps: incremental finite-state parsing and extraction of subject/verb and object/verb relations. Our incremental and cautious approach during the first phase allows the system to deal successfully with complex phenomena such as embeddings, coordination of VPs and NPs or non-standard word order. The extraction requires no subcategorisation information. It relies on POS information only. After describing the two steps, we give the results of an evaluation on various types of unrestricted corpora. Precision is around 90-97% for subjects (84-88% for objects) and recall around 86-92% for subjects (80-90% for objects). We also provide some error analysis; in particular, we evaluate the impact of POS tagging errors on subject/object dependency extraction.

1 Introduction

Dependency extraction from large corpora is mainly used in two major directions: automatic acquisition of lexical patterns [Brent, 1991; Grishman and Sterling, 1992; Briscoe and Carrol, 1994; Sanfilippo, 1994]¹ or for end-user applications such as document indexing or information retrieval [Grefenstette, 1994].

We describe and evaluate an approach for fast automatic recognition and extraction of subject and object dependency relations from large French corpora, using a sequence of finite-state transducers. The extraction is based on robust shallow parsing. We extract syntactic relations without producing complete parse trees in the

traditional sense. The extraction requires no subcategorisation information. It relies on POS information only.

The extraction is performed in two major steps:

1. incremental finite-state parsing annotates the input string with syntactic markings;
2. the annotated string is transduced in order to extract subject/verb and object/verb relations.

Our incremental and cautious approach during the first phase allows the system to deal successfully with complex phenomena such as embeddings, coordination of VPs and NPs or non-standard word order.

We evaluated subject and object dependency extraction on various types of unrestricted corpora. Precision is around 90-97% for subjects (84-88% for objects) and recall around 86-92% for subjects (80-90% for objects).

The paper also provides some error analysis; in particular, we evaluate the impact of POS tagging errors on the extraction process.

2 Incremental Finite-State Parsing

[Aït-Mokhtar and Chanod, 1997] fully describes the incremental finite-state parser. Unlike previous work in this area [Abney, 1991; Roche, 1993; Appelt *et al.*, 1993; Koskenniemi *et al.*, 1992; Voutilainen and Tapanainen, 1993; Chanod and Tapanainen, 1996] the parser combines constructivist and reductionist approaches, in order to maximise efficiency and robustness.

Before the parsing proper, we perform tokenisation, lexical lookup and part-of-speech disambiguation, using the French Xerox tagger [Chanod and Tapanainen, 1995]. The input to the parser is a tagged string represented by a sequence of word-form + tag pairs of the type:

```
le bon vin (the good wine)
<le+DET-SG>
<bon+ADJ-SG>
<vin+NOUN-SG>
```

¹ See also the SPARKLE European project,
<http://www.ilc.pi.cnr.it/sparkle.html>.

The parser output is a shallow parse where phrasal and clausal constructs are bracketed and more or less richly annotated as in:

Jean aime le bon vin (lit.: John likes the good wine)
 [VC [NP Jean NP]/SUBJ :v aime v: VC]
 [NP le [AP bon AP] vin NP]/OBJ

The parser consists of a sequence of transducers. These transducers are compiled from regular expressions that mainly involve the longest match replace operator² [Karttunen, 1996]. Each of these transducers adds syntactic information represented by reserved symbols (annotations), such as segment names, boundaries and tags for syntactic functions. At any given stage of the sequence, the input and the output represent the whole sentence as an annotated string. The output of a transducer is the input to the next transducer.

The parsing process is incremental in the sense that the linguistic description attached to a given transducer in the sequence:

- adds information relying on the preceding sequence of transductions;
- applies only to some instances of a given linguistic phenomenon;
- may be revised at a later stage.

Each step defines a syntactic construction using two major operations: segmentation and syntactic marking. Segmentation consists in bracketing and labeling adjacent elements belonging to a same partial construction (e.g. a nominal or a verbal phrase, or a more partial syntactic chain if necessary). Segmentation includes also the identification of clause boundaries. Syntactic marking annotates segments with syntactic functions (e.g. subject, object, PPObj).

The two operations of segmentation and syntactic marking are performed throughout the cascade in an interrelated fashion. Some segmentations depend on previous syntactic marking and vice versa.

If the constraints encoded by a given transducer do not hold, the string remains unchanged. This ensures that there is always an output string at the end of the cascade, with possibly under-specified segments.

The additional information provided at each stage in the sequence is instrumental in the definition of the later stages of the cascade. Networks are ordered in such a way that the easiest tasks are addressed first. They can be performed accurately with less background information.

² In brief, the expression: $A @-> B \parallel Left _ Right$ indicates that the longest string that matches the regular expression A is replaced by B, if it appears in the context defined by Left and Right.

2.1 Primary Segmentation

A segment (or chunk) is a continuous sequence of words that are syntactically linked to each other or to a governing head (see [Federici *et al.*, 1996; Aït-Mokhtar and Chanod, 1997] for a more detailed description).

Segments are marked using regular expressions such as the following one for NPs:

```
[TBeginNP ~$[TEndNP] TEndNP ]
@->
[NP ... NP]
```

where the longest-match left-to-right Replace Operator (noted @->) inserts [NP and NP] boundaries around the longest sequence that begins with a potential NP start (TBeginNP) and ends with the first NP end (TEndNP) to the right.

In the primary segmentation step, we mark segment boundaries within sentences as shown below. Here NP stands for Noun Phrase, PP for Preposition Phrase and VC for Verb Chunk (a VC contains at least one verb and possibly some of its arguments and modifiers).

Example:

```
[VC
  [VC Lorsqu' [NP on NP] tourne VC]
  [NP le commutateur NP]
  [PP de démarrage PP]
  [PP sur la position PP] [AP auxiliaire AP] ,
  [NP l' aiguille NP] retourne
  VC]
alors [PP à zéro PP] ./SENT3
```

All the words within a segment, except the head, are linked to words in the same segment at the same level. The main purpose of marking segments is therefore to constrain the linguistic space that determines the syntactic function of a word.

As one can see from the example above, segmentation is very cautious. Structural ambiguity inherent to modifiers attachment, verb arguments and coordination is not resolved at this stage.

2.2 Marking Syntactic Functions

Syntactic functions within nonrecursive segments (AP, NP and PP⁴) are handled first because they are easier to mark. The other functions within verb segments and at sentence level (subject, object, verb modifier, etc.) are considered later. The string marked with segmentation and syntactic annotations is the input to the extraction component described below.

³ In English: Turning the starter switch to the auxiliary position, the pointer will then return to zero.

⁴ As a consequence of the cautious segmentation such segments are always nonrecursive.

Extracting dependency relations from annotated strings is not straightforward, as the annotations do not explicitly relate arguments to their governing heads. For instance, shared arguments of coordinated verbs or dependencies across embedded structures need to be resolved during the extraction phase, as illustrated in the example below⁵:

La ville de Lattes rejette toujours la proposition de remonter le niveau du fleuve pour faciliter la circulation des bateaux et n'exclut pas a priori l'idée d'instaurer un péage.

The parser outputs is:

```
[VC [NP La ville NP]/SUBJ [PP de Lattes PP]
:v rejette v: VC] toujours
[NP la proposition NP]/OBJ
[VC de remonter VC] [NP le niveau NP]/OBJ
[PP du fleuve PP] [VC pour faciliter VC]
[NP la circulation NP]/OBJ
[PP des bateaux PP]
[VC et :v n' exclut v: VC] pas a priori
[NP l' idée NP]/OBJ [VC d' instaurer VC]
[NP un péage NP]/OBJ .
```

As objects are marked outside the verb chunks (VC), the extraction phase must still expand the VCs in order to identify object dependencies. Moreover, the extraction must identify *la ville* as the shared subject of the coordinated verbs *rejette* and *exclut*, which are separated by two infinitival clauses. After the preliminary parsing stage, the two coordinated verbs still belong to two different VCs.

3 Subject and Object Recognition and Extraction

The task consists in recognizing the subject and object segments and extracting them along with their respective verbs. Function tagging is performed during the shallow parsing process; then a special transducer recognizes subject/verb and object/verb dependencies (i.e. it finds out which is the subject or object of which verb) and extracts them.

3.1 Subject and object tagging

Potential subjects are marked first. An NP is a potential subject if and only if it is followed by a finite verb and it satisfies some typographical conditions (it should not be separated from the verb with only one comma, etc.). This prevents the NP *Jacques Boutet* for instance from being marked as a subject in the sentence below:

⁵ In English The city of Lattes still rejects the proposal to raise the level of the river to facilitate the boat traffic and does not exclude a priori the idea of imposing a toll

```
[VC [NP le président NP]/SUBJ
[PP du CSA PP], [NP Jacques Boutet NP] ,
a décidé VC]
[VC de publier VC]
[NP la profession NP]/OBJ
[PP de foi PP]./SENT
```

If this type of subject does not exist, we look for inverted subjects under the same typographical constraints.

Other constraints are applied later to eliminate some of the potential subject hypotheses. These constraints are mainly syntactic:

1. A potential subject is not a subject if it has no determiner, unless it is a proper noun or it is a coordinated common noun.
2. A potential subject is not a subject if it immediately follows a PP or an NP (i.e. with no connector in between) and is preceded or followed by another potential subject.
3. A potential subject is not a subject if it is followed by another potential subject with no coordination.

The remaining potential subjects are taken to be actual subjects. The whole process of tagging and correcting subject tags consists of a sequence of replace expressions.

Once the subjects are tagged, another transducer performs object tagging with similar steps and constraints but now only non subject NPs are considered.

3.2 Embeddings and regular expressions

To make this approach work properly, we must be very careful to take into account embedded contexts (embedded clauses, text within parentheses, and so on). For instance, subject constraint 3 above should not apply if the two potential subjects are not at the same level, like the subjects [NP l'usine NP] and [NP le ministre NP] in the following sentence:

```
[VC [NP L' usine NP]/SUBJ ,
[VC que [NP le ministre NP]/SUBJ :v devrait
v: VC]
[VC implanter VC] [PP à Eloyes PP] ,
:v représente v: VC] [NP un investissement
NP]/OBJ
[PP d'environ 148 milliards NP]/N
[PP de francs PP] ./SENT6
```

This also applies throughout the dependency extraction process as described in the next section.

In order to handle this difficulty properly with the finite state calculus, we take advantage of the verb segmenting marks produced by the shallow parser and define a maximal embedding level:

⁶The factory which the minister should establish at Eloyes represents approximately a 148 billion francs investment

level0 = ~\$[BeginVC | EndVC]

level1 = [level0 | [BeginVC level0 EndVC]]*

curlev = [level1 | [BeginVC level1 EndVC]]*

The regular expression *level0* matches any string which does not contain any *BeginVC* or *EndVC* marks, i.e. an embedded clause. *level1* matches any string that contains strings matching *level0* or entire embedded clauses that contains only *level0* matching strings. Therefore, *curlev* (which stands for Current Level) matches strings that either do not contain any embedded clauses, or may contain entire embedded clauses to a maximal depth of 2 embedded levels.

We can easily extend the definition of *curlev* to handle punctuated embeddings such as texts within parentheses or hyphens. We define *Bs* as the beginning of such embeddings (either an opening parenthesis or a hyphen) and *Es* as the ending (a closing parenthesis or another hyphen):

Bs = ["(" | "-"]

Es = [")" | "-"]

and redefine the *curlev* expression as:

level0 = ~\$[BeginVC | EndVC | Bs | Es]

level1 = [level0 | [BeginVC level0 EndVC] | [Bs level0 Es]]*

curlev = [level1 | [BeginVC level1 EndVC] | [Bs level1 Es]]*

Given these definitions, we can control the linguistic space in which rules and constraints apply. For instance, Subject constraint 3 can be written this way:

```
TSUBJ -> [] || -
[curlev
 \[COORD | BeginVC | EndVC | Bs | Es]
 BeginNP ~$EndNP EndNP TSUBJ
 ~[ curlev TSUBJ curlev ] Beginv ]
```

which means: remove a potential subject tag (TSUBJ) whenever there is another potential subject on the right which is not preceded by a coordination. The other potential subject should be the last one before the finite verb. All these considerations are stated over the same sentence level (expressed as *curlev*) so that embeddings relative to a given level and what they contain are not concerned at all.

3.3 Extraction of Dependency Relations

Once the subjects and objects are tagged, we get a shallow parse of the sentence where phrasal units and function tags appears, as shown in the example below:

```
[VC [NP Le président NP]/SUBJ [PP du CSA
PP], [NP Jacques Boutet NP] , a décidé VC]
[VC de publier VC] [NP la profession
NP]/OBJ [PP de foi PP] ./SENT
```

One nontrivial task consists in associating the subjects and objects with the right verb. There are two main difficulties: with subjects and objects of embedded sentences, and with coordinated verbs and shared subjects and objects.

To find out which is the subject/object of a given verb, only the Nps tagged as subjects/objects that are in the same level should be considered. This is necessary in order to avoid overgeneration of dependencies, that is, extracting the subject of an embedded sentence as the subject of the main verb or of another embedded sentence. The definition of *curlev* is very useful for such task. For instance, the transducer obtained from the following regular expression⁷:

```
[
 [ 0 .x. ["SUBJ:"] ]
 [ {?*} .x. 0 ]
 [ BeginNP ~$[EndNP] EndNP [SUBJ : "→" ]
 [ curlev .x. 0 ]
 [Beginv .x. BeginVC] [~$Endv] [Endv .x.
 EndVC]
 [ {?*} .x. 0 ]
 ]
```

can be applied on the previous sentence to extract normal subject/verb dependencies:

```
SUBJ: [NP le président NP] →
      [VC a décidé VC]
```

Other similar expressions are written for the extraction of inverted subjects, shared subjects and objects. All the extracting expressions are joined using the Union operator | and compiled into a single transducer which takes as input shallow parses and produces subject/verb and object/verb dependencies.

3.4 Samples

Below are some samples from the output of the extracting transducer.

⁷ This expression applies to the parsed sentence. It leaves unchanged any NP marked as SUBJ and the corresponding VC found at the same level. All other segments are erased and relational markings (the "SUBJ:" tag and the arrow "→") are inserted.

Sentence 1: A l'interprétation des sentiments présidentiels s'ajoute l'atmosphère de surenchère politique qui précède tout congrès du Parti socialiste.

Shallow parse:

[VC [PP A l' interprétation PP]
 [PP des sentiments PP] [AP présidentiels AP]
 :v s' ajoute v: VC]
 [NP l'atmosphère NP]/<SUBJ
 [PP de surenchère PP] [AP politique AP]
 [VC [NP qui NP]/SUBJ :v précède v: VC] tout
 [NP congrès NP]/OBJ [PP du Parti PP] [AP
 socialiste AP] .

Dependencies:

SUBJ: [VC s' ajoute VC] ←
 [NP l' atmosphère NP]

SUBJ: [NP qui NP] → [VC précède VC]

OBJ: [VC précède VC] ← [NP congrès NP]

Sentence 2: En relançant ainsi délibérément l'agitation sur des enjeux de la grève de septembre 1988, Mr Guillaume met en difficulté le gouvernement et force les autres PDG de l'audiovisuel public à choisir leur camp .

Shallow parse:

[VC [VC En relançant VC] ainsi délibérément
 [NP l' agitation NP]/OBJ
 [PP sur des enjeux PP] [PP de la grève PP]
 [PP de septembre PP] [NP 1988 NP]/N ,
 [NP Mr Guillaume NP]/SUBJ :v met v: VC]
 [PP en difficulté PP]
 [NP le gouvernement NP]/OBJ
 [VC et :v force v: VC]
 [NP les [AP autres AP] PDG NP]/OBJ
 [PP de l'audiovisuel PP] [AP public AP]
 [VC à choisir VC] [NP leur camp NP]/OBJ .

Dependencies:

SUBJ: [NP Mr Guillaume NP] → [VC met VC]

SUBJ: [NP Mr Guillaume NP] → [VC force VC]

OBJ: [VC En relançant VC] ← [NP l' agitation NP]

OBJ: [VC met VC] ← [NP le gouvernement NP]

OBJ: [VC force VC] ←
 [NP les [AP autres AP] PDG NP]

OBJ: [VC à choisir VC] ← [NP leur camp NP]

Sentence 3: Mais, déjà, l'idée que ceux-ci puissent avoir la responsabilité d' un musée classé (établissement dépendant juridiquement d'une ville ou d' un département, mais dont les responsables sont nommés par l'Etat) provoque une certaine émotion chez les nationaux (voir le Monde du 23 novembre 1989).

Shallow parse:

[VC Mais , déjà , [NP l'idée NP]/SUBJ
 [VC que [NP ceux-ci NP]/SUBJ
 :v puissent v: VC]
 [VC avoir VC] [NP la responsabilité NP]/OBJ
 [PP d' un musée PP] [AP classé AP]
 ([NP établissement NP]/N
 [AP dépendant AP] juridiquement
 [PP d' une ville PP]
 ou [PP d' un département PP], mais
 [VC dont [NP les responsables NP]/SUBJ
 :v sont nommés v: VC] [PP par l' Etat PP])
 :v provoque v: VC]
 [NP une [AP certaine AP] émotion NP]/OBJ
 [PP chez les nationaux PP]
 ([VC voir VC] [NP le Monde NP]/OBJ
 [PP du 23 novembre PP] [NP 1989 NP]/N) .

Dependencies:

SUBJ: [NP l' idée NP] → [VC provoque VC]

SUBJ: [NP ceux-ci NP] → [VC puissent VC]

SUBJ: [NP les responsables NP] →
 [VC sont nommés VC]

OBJ: [VC avoir VC] ← [NP la responsabilité NP]

OBJ: [VC provoque VC] ←
 [NP une [AP certaine AP] émotion NP]

OBJ: [VC voir VC] ← [NP le Monde NP]

4 Evaluation

The 15 networks of the parser and extractor require about 750 KBytes of disk space. The speed of analysis is around 150 words per second on a SPARCstation 10, including preprocessing (tokenisation, lexical lookup, POS tagging), parsing and dependency extraction⁸.

We evaluated the extraction on three different types of corpus: newspaper Le Monde (187 sentences, 4560 words, average 24.3 words/sentence), financial reports (524 sentences, 12551 words, average 24 words/sentence) and technical documentation (294 sentences, 5300 words, average 18 words/sentence). The sentences were selected randomly; they are independent from the corpus used for grammar development.

The evaluation concentrated on recognition of surface nominal and pronominal subjects and on nominal direct object. Relations are counted as correct only if both the governing head and its argument are correct (for instance, we counted the subject/Verb relation as erroneous if the verb group was not fully correct, even if the subject was).

Results are given in tables 1 and 2.

⁸ Future work includes optimising the implementation to speed up the process.

Corpus	Actual Subj.	Tagged as Subject	Correct Subj Tag	Precision %	Recall %
Newspaper	350	324	306	94.4	87.4
Financial reports	452	430	418	97.2	92.5
Technic Doc	275	263	238	90.5	86.5

Table 1: Subject recognition on three different corpora

Corpus	Actual Objects	Tagged as Object	Correct Obj Tag	Precision %	Recall %
Newspaper	214	198	171	86.3	79.9
Financial reports	135	160	124	84.4	91.8
Technic. Doc	337	311	275	88.4	81.6

Table 2: Object recognition on three different corpora

4.1 Error analysis

We conducted a detailed error analysis on the newspaper corpus. The major source of subject/verb and object/verb errors was identified for each sentence. If the same source of error accounts for more than one error in a given sentence, it is counted once. There may be also more than one source of error for the same sentence. We identified 81 sources of errors, distributed across the 68 sentences that included at least 1 error (table 3).

Source of errors	Number of occurrences
errors due to tagger	20
errors due to coordination	9
errors due to apposition or NP enumeration	9
errors due to time expressions	7
errors due to pronouns	7
errors due to <i>incise</i>	2
errors due to subject inversion (not <i>incise</i>)	3
messed input (typos, missing blanks)	3
numerical expressions	3
punctuation	3
NP adverbials (other than time expressions)	2
errors due to missing determiners	2
errors due to frozen expressions	2
errors due to clause boundaries	2
errors due to negation (e.g. <i>pas moins de</i>)	2
predicate of non-finite verbs	2
bugs	1
emphatic constructions	1
book title (as object)	1

Table 3. Error Analysis

Among the errors, some can be corrected easily with limited additional work. For instance, the tested version of the parser did not include preprocessing for time adverbials or numerical expressions. The tagset associated with the POS tagger did not provide enough distinctions between different types of pronouns⁹.

The most significant errors are due to long NP sequences (coordination, enumeration, apposition) and to the POS tagger. NP sequences are challenging, as it is difficult for the parser to distinguish between appositions, enumeration and coordinated NPs. Various parameters are to be taken into account (determiners, proper nouns, punctuation, semantic relations, etc.) many of which go beyond the scope of a shallow (or even non shallow) syntactic parser.

4.2 Errors due to POS tagger

There is little to be found in the literature on the impact of POS tagging errors on syntactic analysis. Out of 187 sentences in our newspaper corpus, 20 had subject or object errors due primarily to tagging errors. This does not mean that other sentences did not have tagging errors, but such errors had no impact on the assignment of subjects and objects (neither on the identification of head verbs). This is encouraging, because taggers with 97% accuracy at word level may have a limited accuracy at sentence level (70%).

Table 4 shows the extraction results from sentences where errors due to the POS tagger had no impact. The figures between parentheses refer to the results obtained on the whole corpus, i.e. including tagging errors.

Corpus	Actual	Tagged as	Correct Tag	Precision %	Recall %
Subjects	296	278	266	95.6 (94.4)	89.8 (87.4)
Objects	177	175	152	86.8 (86.3)	85.8 (79.9)

Table 4. Subject and object recognition without POS tagging errors

The improvement is mostly for recall of objects. This is due to the French specific ambiguity for the words *de*, *du*, *des* which can be indefinite determiners or preposition+determiner sequences. The tagger does not do a good job at distinguishing between the two tags, especially in postverbal position. This has an obvious impact on the dependency analysis, as segments mistagged as PPs cannot be identified as subjects or objects.

⁹ This project started around mid-96; some of the obvious improvements are already part of a recently updated version.

5 Conclusion

We presented a finite-state approach to subject and object dependency extraction from unrestricted French corpora. The method involves incremental finite-state parsing and transductions for dependency extraction, using only part-of-speech information. The evaluation on different corpora (22,000 words) shows high accuracy and precision.

The method is being expanded to other dependency relations and to other languages.

The extracted dependencies can be used to automatically build subcategorisation frames or semantic restrictions for verb arguments, which, in turn, can be reused in further steps of incremental parsing. They can also be used for knowledge extraction from large corpora and for document indexing.

Acknowledgments:

We are grateful to Annie Zaenen and Lauri Karttunen for their editorial advices.

References

[Abney, 1991] Steve Abney. Parsing by chunks. *Principled-Based Parsing*, R. Berwick, S. Abney, and C. Tenny, editors. Kluwer Academic Publishers, Dordrecht, 1991.

[Aït-Mokhtar and Chanod, 1997] Salah Aït-Mokhtar and Jean-Pierre Chanod. Incremental Finite-State Parsing. *Proceedings of ANLP-97*, Washington, 1997.

[Appelt *et al.*, 1993] Douglas E. Appelt, Jerry R. Hobbs, John Bear, David Israel, and Mabry Tyson. FASTUS: A Finite-State Processor for Information Extraction from Real-World Text. *Proceedings IJCAI-93*, Chambery, France, August 1993.

[Brent, 1991] Michael Brent. Automatic Acquisition of subcategorization frames from untagged texts. *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*. Berkeley, CA, 1991.

[Briscoe and Carroll, 1994] Edward Briscoe and John Carroll. Towards Automatic Extraction of Argument Structure from Corpora. Technical Report MLTT-006, Rank Xerox Research Centre, Meylan, 1994.

[Chanod and Tapanainen, 1995] Jean-Pierre Chanod and Pasi Tapanainen. Tagging French - comparing a statistical and a constraint-based method. *Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics*, pages 149--156, Dublin, 1995.

[Chanod and Tapanainen, 1996] Jean-Pierre Chanod and Pasi Tapanainen. A Robust Finite-State Parser for French.

Proceedings of ESSLLI'96 Workshop on Robust Parsing Prague, 1996.

[Federici *et al.*, 1996] Stefano Federici, Simonetta Montemagni and Vito Pirrelli. Shallow Parsing and Text Chunking: a View on Underspecification in Syntax. *Proceedings SLLI'96 Workshop on Robust Parsing*, Prague, 1996.

[Grefenstette, 1994] Gregory Grefenstette. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Press, Boston, 1994.

[Grishman and Sterling, 1992] Ralph Grishman and John Sterling. Acquisition of Selectional Patterns. *Proceedings of Coling-92* Nantes, 1992.

[Karttunen, 1996] Lauri Karttunen. Directed replacement. *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, Santa Cruz, USA, Association for Computational Linguistics, 1996.

[Koskenniemi *et al.*, 1992] Kimmo Koskenniemi, Pasi Tapanainen, and Atro Voutilainen. Compiling and using finite-state syntactic rules. *Proceedings of the Fourteenth International Conference on Computational Linguistics COLING-92*, Nantes, 1992.

[Roche, 1993] Emmanuel Roche. Analyse syntaxique transformationnelle du français par transducteurs et lexique-grammaire, Ph.D. dissertation, Université de Paris 7, 1993.

[Sanfilippo, 1994] Antonio Sanfilippo. Word Knowledge Acquisition, Lexicon Construction and Dictionary Compilation. *Proceedings of Coling-94*, Kyoto, 1994.

[Voutilainen and Tapanainen, 1993] Atro Voutilainen and Pasi Tapanainen. Ambiguity resolution in a reductionistic parser. *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics*, Utrecht, 1993.