Statistical Ambiguity Resolution for Principle-Based Parsing

Cathy Berthouzoz and **Paola Merlo***
Laboratoire d'Analyse et de Technologie du Langage
Linguistic Department - University of Geneva
{berthouzoz,merlo}@latl.unige.ch

Abstract

The goal of syntactic ambiguity resolution is to produce the naturally preferred interpretation of any given input, while limiting computational resources. We present in this paper a statistical pruning device developed for a principle-based parser which attempts to satisfy this goal. We compare the performances of the resulting hybrid parser with an instance of the same parser including a symbolic pruning device, showing that the statistical approach has better performance.

1 Introduction

Differently from programming languages, natural languages are non-deterministic, since they are ambiguous. When dealing with natural language processing one of the main tasks is to resolve ambiguity at the lexical, syntactic, semantic and pragmatic levels in order to produce naturally preferred analyses, *i.e.* preferred interpretations, of any given input, while limiting computational resources.

Beside the inherent ambiguity of the language, the form of the underlying grammar may add what is called *parasitic ambiguity*, ambiguity resulting from the underspecification of the grammar. This problem is particularly acute for domain-independent, wide-coverage grammars which tend to be expressed at a very abstract level of representation.

Full parallelism – all the alternatives are processed from any point of ambiguity – is one way of simulating non-determinism. This approach is computationally expensive as it produces a large amount of irrelevant analyses. To avoid the combinatorial explosion of alternatives that full parallelism generates, limited non-deterministic parsing algorithms have been elaborated (see for review Dowty *et al.* 85; Reyle & Rohrer 88) which use a pruning device to limit the number of hypotheses that are pursued at any point of ambiguity. Pruning devices may be based on symbolic information – the preferences are an effect of structural complexity or unsatisfied grammatical functions (Gibson 91; Stevenson 94), or on statistical information – preferences are the result of frequency-based activation of prestored representations (MacDonald 94; Jurafsky 96; Ratnaparkhi 97). On the one hand, the preferences are stated according to structure-based heuristics (Walther 93), on the other hand they are stated according to probabilities estimated by corpus-based frequencies (Magerman & Marcus 91; Briscoe & Carroll 93; Bod 95; Collins 96).

Until recently, symbolic pruning devices were used with symbolic parsers – rich structural descriptions generated according to a symbolic grammar– while statistical pruning devices were used with statistical parsers – raw descriptions generated according to a statistical language model. A new promising area is emerging: hybrid parsers, which combine statistical model of preferences with rich structural descriptions (Srinivas *et al.* 95; Tugwell 95).

The IPS parser developed at LATL (Wehrli 92; Laenzlinger & Wehrli 93 among others) uses an abstract, domain-independent, principle-based (*GB-style*) grammar. Since the grammar is underconstrained, the parser generates a large amount of parasitic hypotheses, namely it generates many types of temporary ambiguities even for unambiguous sentences. This parser constitutes therefore a good test bed for filtering devices.

In this paper we present the statistical pruning device we have developed for the IPS parser: a beam search filter based on probabilities of the different analyses. We compare the performances of the resulting hybrid parser with a previous version of the parser that used structural costs to

* We gratefully acknowledge the support of the Swiss National Science Foundation, for funding this research under grant 12-043283.95. The research was completed while the second author was a visiting scientist at the University of Pennsylvania, supported jointly by the Swiss National Science Foundation advanced researcher fellowship 8210-46569 and by the Institute for Research in Cognitive Science.

prune unpreferred analyses.

2 Materials and Method

2.1 The Corpus

A test suite of 61 sentences was created by collating three different sorts of sentences: 20 sentences are instances of different types of PP-attachment ambiguity, ranging from ambiguity between an NP/VP attachment, to different NP attachment sites. Testing performance on this kind of ambiguity is important, as different PPs can attach at different sites, generating a large number of alternative analyses (Church & Patil 82). Moreover, 20 sentences reflecting the so called NP/S ambiguity were selected from the annotated version of the Wall Street Journal. 10 verbs were selected by stratified sampling, in such a way that 5 verbs were biased for an NP complement and 5 verbs were biased for an S complement, and that the differential in preference for the 2 continuations was strong. Two sentences were adapted from the text, one containing an NP resolution and one containing an S resolution.[1] Finally, 20 miscellaneous sentences from a previous test suite were added, in order to test that filtering to handle specific kinds of ambiguity would not decrease performance on other constructions.

2.2 Performance Measures

In general, a good way of establishing the performance of a parsing system consists in determining how much the output of the system differs from a previously established gold standard. This is usually done by measuring *precision* and *recall*. Precision is a measure of accuracy, which tells what percentage of the total output is actually correct. Recall is a measure of coverage, telling how many of the correct analyses were actually recovered by the parsing system.

We define two different units on which to calculate precision, so that these measures are illustrative of the outputs and the architecture of our parser. First, in order to asses the complexity of the algorithms, and the amount of work the parser performs, we use a measure that takes all completed analyses into account, called *absolute precision*. The second measure derives from the fact that the IPS parser can relax constraints during a parse to achieve better robustness. Consequently, many of the completed parses are in fact labelled as "grammatically incorrect" by the parser. It is therefore meaningful to establish the measures of precision based only on the parses that the parser labels as grammatically correct, called *relative precision*. We believe that this is the measure that most closely resembles the factors that are taken into account into precision and recall evaluation of other parsers presented in the literature.

We define two indices of performance: an index of global performance, based on all correct analyses, and an index of the performance in establishing preferences, that takes only the preferred analysis into account.[2]

2.3 The Parsing Architecture

The IPS parser is a chart parser (Kay 86), which pursues all alternatives in parallel. Differently from most chart parsers however, it does not use a grammar of context-free rules to build the structure, but rather it proceeds as a licensing parser (Abney 89), where only three types of actions are supported: projection, attachment to the left (specifiers) and attachment to the right. The advantage of using a licensing technique is that the parser is data driven and that it is incremental, as incoming tokens are immediately incorporated in the structure. Alternatives are considered concurrently, and a small number of heuristics are used to restrict the hypothesis set. For instance, satisfaction of selectional features is favoured, thus, if an incoming verb form can be attached either

[1] This ambiguity is going to provide us with the data necessary to compare our results with the various experimental results proposed in the psycholinguistic literature, where this ambiguity has been extensively studied (Trueswell et al. 93). The classical landmark *While she was mending the sock fell off her lap* was added too.

[2] Thus, we have six indices:
global-absolute-precision = Ncorrect/Ncomplete
preferred-abs-precision = Npreferred/Ncomplete
global-relative-precision = Ncorrect/Ngrammatical
preferred-rel-precision = Npreferred/Ngrammatical
global-recall = Ncorrect/Ntotal-expected
preferred-recall = Npreferred/Npreferred-expected
where
Ncomplete = number of complete analyses built by the parser.
Ncorrect = number of complete analyses judged correct by the authors.
Ngrammatical = number of complete analyses that do not include any grammatical violation, according to the parser.
Ntotal-expected = number of correct expected analyses that the parser should produce, as judged a priori by the authors.
Npreferred = number of preferred analyses, in our case one for each sentence.
Npreferred-expected = number of preferred analyses that the parser should produce.

	Total	Preferred
Absolute Precision	0.090	0.047
Relative Precision	0.328	0.160
Recall	0.916	0.983

Table 1: Performance measures of the system without any filtering

as a complement to an auxiliary or as a main verb, preference will be given to the former. This set of heuristics is going to be replaced by the structural and statistical measures under discussion. Although still under development, the IPS parser, which uses a lexical database exceeding 80,000 entries, has a fairly broad grammatical coverage including simple and complex sentences, complex determiners and possessives, yes/no and *wh*-interrogatives, relatives, passive, some comparatives as well as some cases of coordination (of same category).

In order to have a clear base line, we run the IPS parser on the test sentences. Precision and recall of the system without any filtering device is given in Table 1[3]. This table shows that the system without any filtering device recovers more than 91% of all the possible interpretations for the given input (total recall of 0.916) and almost 100% of the preferred interpretations (preferred recall of 0.983). But in order to achieve these results, it produces 90% of irrelevant analyses in one case and 95% in the second case (absolute total precision of 0.090 and 0.047 respectively).

3 Structural Beam Search

The IPS system integrates a filtering device to prune the generated hypotheses in the course of parsing, and a ranking device to score the complete analyses of a given sentence and retrieve the most preferred one at the end of the parse.

A structural beam search device can be defined by assigning costs to all tree construction operations. The costs we used to define the total cost of an analysis where based on empirical psycholinguistic evidence and on empirical testing, as described in (Berthouzoz & Merlo 97). The structural cost will serve as the metric for the beam search algorithm developed here: at any step of the parse, only the constituents with a cost lying within a given bound from the lowest cost will be kept for further processing. The beam width can be set as a number of hypotheses – at each level, the N constituents with the lowest costs are kept – or as a threshold – at each level, all the constituents with a cost between $cost_{min}$ and $cost_{min} + \alpha$ are kept, where α is an arbitrary cost. We have adopted the latter option.

Width	Abs-Prec	Rel-Prec	Recall
Total			
b60	0.207	0.525	0.706
b80	0.169	0.461	0.849
b100	0.127	0.407	0.866
b120	0.107	0.369	0.874
b140	0.097	0.347	0.874
Preferred			
b60	0.123	0.313	0.862
b80	0.089	0.242	0.914
b100	0.065	0.209	0.914
b120	0.056	0.191	0.931
b140	0.050	0.180	0.931

Table 2: Precision and recall of IPS with a structure-based beam filtering device of width 60, 80, 100, 120 and 140

Several experiments were run to determine the best width of the beam, as shown in Table 2. We retain the width of 60 for all further comparisons, because it achieves the best absolute precision.

4 Statistical Beam Filtering

To overcome the main drawbacks of pure symbolic or statistical methods, hybrid approaches have recently emerged in parsing (Bod 95; Tugwell 95; Voutilainen & Padró 97; Chen & Chen 94). To take advantage of both approaches by integrating statistics into a symbolic parser, we substitute the structure-based selection device of the parser, described above, with a frequency-based mechanism, which we describe and evaluate in the following sections. A hybrid approach to parsing

[3]The number of total expected analyses is the number of interpretations one sentence has, that means one for non ambiguous sentences such as in *While John was mending the sock fell off his lap*, more than one for ambiguous sentences such as *He saw a man with the telescope*, which has two possible interpretations.

This is a much stricter test than what is usually done, as it is based on the assumption that humans would have 100% agreement on what constitute the best analysis for a sentence, if they were asked to classify the output of the parser. We know that humans do not show perfect agreement when asked to perform this task, although currently no precise measure of inter-judge agreement is available.

consists of two components: a grammar, which defines the event space, and a probabilistic model. We describe the two components in turn.

4.1 Defining dependencies

In the representation used by the IPS parser, possible sentence configurations are determined by a modified version of the $\overline{X}$ schema (Chomsky 86), in which the bar level is not explicitly defined.

XP = Specifiers X Complements

for X $\in$ { N, V, Adj, Adv, D, P, Conj, T, C, F } the set of possible heads.

The richness of the grammatical relations are stated in terms of attachment types in this quite flat structure. The parser discriminates 5 types of attachment of full categories (adjuncts, functional complements, subcategorized complements, specifiers, and noun-modifiers) and 3 types of attachment of empty categories (specifier, adjunct and subcategorized complements traces). In order to limit overgeneration, the grammar defines constraints over XP, Specifiers and Complements. Abstractly, the grammatical model used by the parser enables us to represent the parse tree T by a set of maximal projections X and a set of dependencies D between maximal projections given the sentence S.

A dependency $d_i \in D$ is defined as a relation $r(i)$ holding between two maximal projections, $x_i \in X$, called the *base,* and $x_j \in X$ called the *site*. The relation is represented by a 3-tuple including the two maximal projections x_i and x_j, and the type of the relation, *dep*.

$$r(i) = (x_i, dep, x_j) \quad (1)$$

If the grammatical label is a verb, then the lexical information is also taken into account. Thus, we can define the event space for the statistical model as follows.

$$\Omega = \{X \times D \times X\} \cup \{X \times D \times VERB\} \quad (2)$$

Where X is the set of all maximal projections, D is the set of all grammatical relations, and $VERB$ is the set of all verbs (lexical items).

For instance, in the parse tree of the figure 1 below, we can define the following relations.[4]

$$d_1 = r_1 = (X_1,\ \ spec,\ \ ate)$$
$$d_2 = r_2 = (X_2,\ \ fcompl,\ \ X_1)$$
$$d_3 = r_3 = (X_3,\ \ identity,\ \ ate)$$
$$d_4 = r_4 = (X_4,\ \ subcat,\ \ ate)$$
$$d_5 = r_5 = (X_5,\ \ fcompl,\ \ X_4)$$

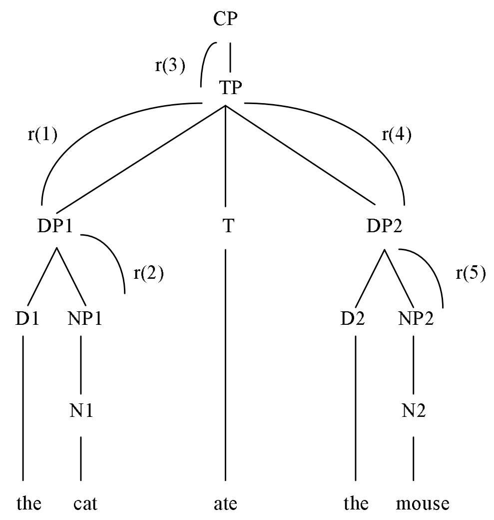

Figure 1: A simple parse tree

4.2 The Probabilistic Model

The statistical model we propose defines the probability of a parse at two different levels of granularity. For the verb, the model takes into account the co-occurrence frequencies of the single lexical items with different subcategorization frames; for all other syntactic categories, the model is defined only at the level of the syntactic dependencies between maximal projections, namely at a level at which lexical content is not taken into account. The reason for adopting this mixed approach is that probabilities defined at the syntactic level are going to be domain-independent, and the ensuing model is going to be more easily portable and useful even across domains. On the other hand, since the verb is the most informative item in the sentence, probabilities that depend on verbs should be defined at a finer level of granularity, so as to be more informative.

We define a syntactic tree T as the union of the sets of syntactic labels X and the set of dependencies D describing the tree. We write T= (D,X). The probability of the syntactic tree T given the sentence S can be defined as follows.

[4] As our grammar allows empty categories, x_i may be empty. Thus the number of triples is not equal to the number of words in the sequence. We can however set the uperbound limit to the number of triples – this is equivalent to the size of a parse tree generated by a Government and Binding theory grammar – as a number linearly related to the length of the input sentence (Correa 91).

$$P(T|S) = P(X, D|S) = P(X|S)P(D|S, X) \quad (3)$$

As there is only one lexical tag associated to each word in the structure, the sentence can be reduced to a set of lexical tags $\{t_1, t_2, \ldots, t_{i-1}, < V, lexeme >, ti+1, \ldots, t_n\}$, with the notable exception of the verb, whose lexical content – lexeme – is associated with the tag V.

Since the grammar is defined in such a way that each lexical tag t projects one and only one maximal projection $x \in X$, so that $P(X|S) = 1$, (3) can be simplified as follows.

$$P(T|S) = P(X, D|S) = P(D|X) \quad (4)$$

The conditional probability of a particular dependency $d \in D$ given a syntactic category $x_i \in X$ is equal to the conditional probability that x_i is involved in a relation $r(i)$ given x_i.

$$P(d|x_i) = P(r(i)|x_i) \quad (5)$$

This model assumes independence of relations, thus the probability of the parse tree is the product of the probabilities of the relations defined on it.

$$P(T|S) = P(D|X) = \prod_{i=1\ldots n} P(r(i)|x_i) \quad (6)$$

4.3 Estimation of probability of events

The basic feature of the probability model is that the event space is defined at two levels of granularity – lexical if the verb is involved, syntactic for all other relations. Thus this model is fundamentally different from other dependency-based statistical parsers which are completely lexicalized, such as (Collins 96; Charniak 95).

If the site x_j is a VP and the dependency *dep* is the subcategorized complement attachment, this means that the base x_i is an argument of the verb. Then, these probabilities are estimated by taking into account lexical elements (the verb). We determine the global distribution of subcategorization frames and the posterior distribution of arguments for any given verb.

In all other cases, probabilities are estimated for dependencies at the syntactic level by determining the global distribution of attachment between syntactic categories.

We condition the probability of D events on the syntactic category of the *base*. For instance, the probabilities of the relations holding on the tree above would be estimated as follows:

$$P(r(1)|x_1 = DP) \approx \frac{C(< DP, spec, ate >)}{C(DP)}$$

$$P(r(2)|x_2 = NP) \approx \frac{C(< NP, fcompl, DP >)}{C(NP)}$$

$$P(r(3)|x_3 = VP) \approx \frac{C(< NP, identity, ate >)}{C(VP)}$$

$$P(r(4)|x_4 = DP, eat) \approx g\left(\frac{C(< DP, subcat, ate >)}{C(DP)}, \frac{C(< eat, [np] >)}{\sum_k C(< eat, frame_k >)}\right)$$

$$P(r(5)|x_5 = NP) \approx \frac{C(< NP, fcompl, DP >)}{C(NP)}$$

g is a composition function, that combines the two sources of information, structural and lexical for the verb-related dependencies[5]. In practice, it is an exclusive OR function, so that lexical information is always used when available, and syntactic information is used only when lexical information is not available.

4.4 Approximating Probabilities by Corpus Frequencies

In order to obtain good approximations of probabilities, the corpus has to be large enough to avoid sparse data, and reflect natural use of language. We are thus using the one million words portion of the 1989 Wall Street Journal contained in the Penn TreeBank (Marcus *et al.* 93) which is available in a tagged and parsed version. The resulting tree structured text of about 50'000 sentences may be searched for patterns specifying node names and relationships between nodes.

The output structures of the corpus differ considerably from the parser output structures. For instance, a parsed sentence in the PTB annotation format is given in (a) while the same sentence parsed by the parser we use is given in (b) below.

[5] The arguments of a verb are put into square brackets, thus [np] means the frame contains one argument which is the direct object of the verb, while $frame_k$ can take values in all the different possible frames ([], [np], [np np], [np pp], [np s], [np sbar], [np ap], [pp], [pp pp], [pp s], [pp sbar], [s], [sbar], [ap]).

Attachment	Maximal Projections	Category Range
Specifier	XP = (YP X)	X ∈ {Adj, Adv, C, D, T, N, V} and Y ∈ {Adj, Adv, C, D, P}
Noun-modifier	XP = (YP X)	X,Y = N
Specifier-trace	XP = (YP X)	X= T, YP= empty DP and ∃ ZP=UP for Z = C, U = D and UP is antecedent of YP
Functional-complement	XP = (X YP)	X ∈ {Adv, C,Conj, D, F, P, T}, Y ∈ {A, C, Conj, D, N, P, T, V} and X selects YP
Subcategorized-complement	XP = (X YP)	X ∈ {Adj, D,F, N, V}, Y ∈ {C, Conj, D, F, P} and X subcategorizes YP
Subcat-compl-trace	XP = (X YP)	X ∈ {Adj, N, P, T, V}, YP ∈ empty DP and ∃ ZP = (UP Z) for Z = C, U ∈ {D,P}, X subcategorizes UP and UP is antecedent of YP
Adjunct	XP = (X YP)	X ∈ {N, V},YP ∈{AdvP,CP}
Adjunct-trace	XP = (X YP)	X ∈ {V, T}, YP ∈ empty AdvP and ∃ ZP = UP for Z = C, UP = AdvP and UP is antecedent of YP

Table 3: Different types of attachments in IPS

1. (S (NP-SBJ They) (VP do not want (S (NP-SBJ (-NONE- *))(VP to (VP see (S (NP-SBJ anything) (VP change)))))))
2. (CP (TP (DP They)i (T' do (VP (ADVP not) (V' want (CP (TP (DP ei) (T' to (VP see (CP (TP (DP (NP anything)) (T'(VP change)))))))))))))

To map the different attachments types into the PTB structures, we describe the site of attachment, XP, which corresponds to the dominating node in the corresponding PTB tree, and the attached constituent, YP, which can be specifier or complement. The constraints include co-occurrence of possible categories and specific features of head and modifiers. Table 3 defines the eight attachment types. In the definitions, the attachment applied between constituent YP, the base, and its context XP, the site. Example mappings are shown in Table 4.

4.5 Results and Discussion

In this section, we present and discuss the results of the statistical beam search. There are two strategies for setting the beam width.[6]

[6] The parser architecture is as described in the previous section, a chart parser that prunes low preferred analyses by using a beam.

The beam width can vary, because it is set as the maximum ratio between the highly-ranked structure and the less probable ones above which a structure is pruned. For instance, if P_h is the probability of the most probable analysis, P_l is the probability of a less probable alternative and α is the beam width, then if $P_l < P_h/\alpha$ the worse structure is pruned. We tried several values from 1000 to 1.2 but we kept 1000, 150 and 20 for further investigation.

A more robust way of using a beam search algorithm consists in using a fixed beam width, which is set as the maximum number N of alternatives to be kept at each step of the parse. We tried 20, 13, 10 and 5 alternatives. This N-best strategy seems more accurate, probably because, even if one analysis is very much preferred, other analyses are kept in the beam, thus recovery from bad estimates of probability is possible.

Table 5 shows that both variable and fixed beam searches perform better than a structurally-based search. However, the fixed beam overall outperforms the variable width beam. We can observe in this table that with a fixed beam width of 20, the parser can recover almost the same amount of interpretations as the parser without any filtering device (total recall of 0.908 and 0.916

Wh-elements *Who* are you? *When* are you leaving?	
IPS: XP = CP and YP $\in$ {AdjP, AdvP, DP, PP} and YP has feature *wh-element.*	
PTB: (SBAR| SBARQ (WHNP| WHADJP| WHPP| WHADVP ...) ...)	
Sentential adverbs *Sometimes* John feels tired.	
IPS: XP = CP and YP = AdvP and YP has feature *sentential* or *standard*	
PTB: (S| SBAR| SQ| SBARQ (AdvP ...) ...)	
Preverbal adverbs He *never* comes. He had *often* thrown the ball.	
IPS: XP $\in$ VP, TP and YP = AdvP and YP has feature *time* or *frequency*	
PTB: (S| SQ ... (ADVP-TMP ...) (VP ...)...), (VP (ADVP-TMP ...) (VB* ...)...)	

Table 4: Examples of mapping functions between the Penn Tree Bank and the representation used in the IPS parser

respectively). The values of the preferred recall (0.793 and 0.983 respectively) tell us however that the preferred interpretation is not the most probable one in 20% of the cases, while it is the first ranked in almost 100% of the cases.

The statistical model is also computationally less expensive than the structural one, as it hypothesizes many fewer constituents than the structural beam parser. In comparing structural beam 60 to variable-width probabilistic 150, which have similar global precision, we found that the former requires 5145 constituents to reach 405 complete analyses at the end (12.7 constituents per analysis), while the latter requires 2853 constituents for 481 analyses (5.93 constituents for each analysis).

4.6 Comparison to other models

The parsing system reported in (Srinivas *et al.* 95) shares several features with the version of IPS presented here. Both parsers make use of a wide-coverage linguistically motivated grammar, although inspired by different formalisms. Moreover they both make use of a filtering component in order to disambiguate the analyses generated by the parser. Although in design and combination of different sources of information the two systems are similar, they show some differences in their more detailed organization. Srinivas *et al.* 95 make use of a complex filtering component, which consists of a combination of three techniques: POS tagging, tree filtering and parse ranking. Moreover, while they combine structure-based heuristics and statistical information, we simply replaced any structural heuristics by statistical information.

Both approaches, however, make use of a database containing composite statistics – our parser uses dependency frequency and subcategorization frequency in the case of verbs, for instance. The use of several sources of frequency information and a separate grammar and statistical component is in opposition to approaches such as those proposed by (Collins 97). In this work, the grammar is induced from the corpus directly, and there is no separation between the symbolic and the statistical component. In other words, there is no discrete information other than what is needed to the definition of the space of events of the probabilistic function. Consequently, Collins's models 2 and 3 are a mean to approximate linguistic knowledge, which is instead given in our model.

5 Conclusion and Future Research

We have presented some experiments in evaluating the performance of a symbolic parser for English (IPS). Due to its massively parallel nature, IPS has good recall on the set of sentences that we tested, but precision is very low, showing that overgeneration of constructions needs to be reduced. We have tested two methods. In the structural method, both the generating and the filtering device are based on knowledge stored in a linguistically-defined grammar. In the hybrid method, the generating device is structure-based but the filtering is done based on statistical information extracted from a corpus. We observe that the statistical method outperforms the structural approach, that it is less computationally expensive, and it is easier to define and develop. Current results on the statistical beam search can be further improved by a better back-off model, for unseen cases of subcategorization frames.

Global								
	s60	vp1000	vp150	vp20	fp20	fp13	fp10	fp5
Absolute Precision	0.207	0.183	0.202	0.274	0.255	0.303	0.325	0.423
Relative Precision	0.525	0.532	0.581	0.669	0.548	0.597	0.616	0.746
Recall	0.706	0.832	0.815	0.664	0.908	0.882	0.824	0.714
Preferred								
Absolute Precision	0.123	0.074	0.083	0.128	0.108	0.133	0.142	0.214
Relative Precision	0.313	0.215	0.240	0.314	0.234	0.261	0.270	0.377
Recall	0.862	0.690	0.690	0.638	0.793	0.793	0.741	0.741

Table 5: Overview of filtering techniques: s60:structural with 60 width; vp1000, vp150, vp20:variable probabilistic beam with width of 1000, 150 and 20; fp20, fp13, fp10, fp5: fixed probabilistic beam with respective widths

References

(Abney 89) Steve Abney. A Computational Model of Human Parsing. *Journal of Psycholinguistic Research*, vol. 18, pp 129-144, 1989.

(Berthouzoz & Merlo 97) Cathy Berthouzoz and Paola Merlo. Filtrage Structurel versus Filtrage Statistique: Expériences sur la Résolution de l'Ambiguïté syntaxique. *Actes des 1ères JST FRANCIL 1997*, pp. 353-359. Avignon, France, 1997.

(Bod 95) Rens Bod. *Enriching Linguistics with Statistics.* PhD thesis, University of Amsterdam, 1995.

(Briscoe & Carroll 93) Edward Briscoe and John Carroll. Generalized probabilistic LR parsing of natural language (corpora) with unification based grammars. *Computational Linguistics*, 19(1):25–60, 1993.

(Charniak 95) Eugene Charniak. Parsing with context-free grammars and word statistics. Technical Report CS-95-28, Department of Computer Science, Brown University, 1995.

(Chen & Chen 94) Huang-hua Chen and Hsin-Hsi Chen. Extracting Noun Phrases from Large-Scale Texts: A Hybrid Approach and Its Automatic Evaluation. *Proceedings of the ACL*, 1994.

(Chomsky 86) Noam Chomsky. *Knowledge of Language: Its Nature, Origin and Use.* Praeger, New York, 1986.

(Church & Patil 82) K. Church and R. Patil. Coping with syntactic ambiguity or how to put the block in the box on the table. *American Journal of Computational Linguistics 8(3-4)*, 139–149, 1982.

(Collins 96) Michael Collins. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the Association for Computational Linguistics*, Santa Cruz, California, 1996.

(Collins 97) Michael Collins. Three Generative, Lexicalised Models for Statistical Parsing. In *Proceedings of the ACL/EACL*, Madrid, Spain, 1997.

(Correa 91) Nelson Correa. Empty Categories, Chain Binding and Parsing. *Principle-Based Parsing: Computation and Psycholinguistics*, Berwick, Abney and Tenny (eds), pp 83-122, 1991.

(Dowty *et al.* 85) D. Dowty, L. Karttunen and A. Zwicky (eds). *Natural language parsing. Psychological, computational, and theoretical perspectives.* Cambridge University Press, 1985.

(Gibson 91) Edward Gibson. *A Computational Theory of Human Linguistic Processing: Memory Limitations and Processing Breakdown.* PhD. Thesis, Carnegie Mellon University, 1991.

(Jurafsky 96) Daniel Jurafsky. A probabilistic model of lexical and syntactic access and disambiguation. *Cognitive Science*, 20(2), 1996.

(Kay 86) Martin Kay. Algorithm Schemata and Data Structures in Syntactic Processing. *Readings in Natural Language Processing*, Sparck Jones and Webber (eds). Morgan Kaufmann, Grosz, 1986.

(Laenzlinger & Wehrli) Christopher Laenzlinger and Eric Wehrli. Théorie linguistique et traitement automatique du langage. *La Traductique*, P. Bouillon et A. Clas (eds), Les Presses de l'Université de Montréal, pp. 199-211, 1993.

(MacDonald 94) Maryellen MacDonald. Probabilistic Constraints and Syntactic Ambiguity Resolution. *Language and Cognitive Processes*, vol. 9(2), pp 157-201, 1994.

(Magerman & Marcus 91) David Magerman and Mitch Marcus. Pearl: A probabilistic parser. In *Proceedings of the European Chapter of the Association for Computational Linguistics*, Berlin, 1991.

(Marcus *et al.* 93) Mitch Marcus, Beatrice Santorini and M.A. Marcinkiewicz. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19:313–330, 1993.

(Ratnaparkhi 97) Adwait Ratnaparkhi. A Linear Observed Time Statistical Parser Based on Maximum Entropy Models. In *Proceedings of the EMNLP-2*, Providence, Rhode Island, August 1-2, 1997.

(Reyle & Rohrer 88) U. Reyle and C. Rohrer (eds). *Natural Language Parsing and Linguistic Theories.* Reidel, 1988.

(Srinivas *et al.* 95) B. Srinivas, C. Doran and S. Kulick. Heuristics and Parse Rankings. *Proceedings of International Workshop on Parsing Technologies*, Prague, 1995.

(Stevenson 94) Suzanne Stevenson. *A Competitive Attachment Model for resolving Syntactic Ambiguities in Natural Language Parsing.* University of Maryland at College Park, 1994.

(Tugwell 95) David Tugwell. A State-Transition Grammar for Data-Oriented Parsing. *Proceeding of the EACL*, 1995.

(Voutilainen & Padró 97) Atro Voutilainen and Lluís Padró. Developing a hybrid NP parser. *5th ANLP*, 1997.

(Walther 93) Catherine Walther. Psycholinguistically-based selection of analyses in IPS and FIPS. Notes techniques 93/3, LATL, University of Geneva, 1993.

(Wehrli 92) Eric Wehrli. The IPS system. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, pages 870–874, Nantes, 1992.