Chapter 4

# EVALUATING PART-OF-SPEECH TAGGING AND PARSING

## *On the Evaluation of Automatic Parsing of Natural Language*

Patrick Paroubek

*Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur*
*LIMSI-CNRS, Orsay, France*

pap@limsi.fr

**Abstract** The aim of this chapter is to introduce the reader to the evaluation of part-of-speech (POS) taggers and parsers. After a presentation of both POS tagging and parsing, describing the tasks and the existing formalisms, we introduce general considerations about evaluation of Natural Language Processing (NLP). Then we raise a point about the issue of input data segmentation into linguistic units, a crucial step in any evaluation related to language processing. We conclude by a review of the current evaluation methodologies and average levels of performance generally achieved for POS tagging and parsing.

**Keywords** Natural language processing; Evaluation; Part-of-speech (POS) tagging; Parsing.

## 1 POS Tagging

Part-of-speech (POS) tagging is the identification of the morphosyntactic class of each word form using lexical and contextual information. Here is how Brill's tagger (Brill, 1995) tags the first sentence of this paragraph. Each line holds respectively: a token number, a word form, a POS tag, and a short tag description.

```
0 part-of-speech tagging VBG  verb, gerund or present participle
1 is VBZ  verb, present tense, 3rd person, singular
2 the DT  determiner
3 identification NN  noun, singular or mass
4 of IN  preposition or subordinating conjunction
```



*L. Dybkjær et al. (eds.), Evaluation of Text and Speech Systems,* 99–124.

```
 5 the DT  determiner
 6 morphosyntactic JJ  adjective
 7 class NN  noun, singular or mass
 8 of IN  preposition or subordinating conjunction
 9 each DT  determiner
10 word NN  noun, singular or mass
11 form NN  noun, singular or mass
12 using VBG  verb, gerund or present participle
13 lexical JJ  adjective
14 and CC  conjunction, coordinating
15 contextual JJ  adjective
16 information NN  noun, singular or mass
```

Brill's tagger uses the Penn Treebank[1] tagset (Marcus et al., 1993). The tagset regroups all the tags used to represent the various word classes. Ideally, a tagset should have the capacity to integrate all the morphosyntactic information present in the lexical descriptions of the words, if any is available. It should also have the capacity to encode the information needed to disambiguate POS tags in context, and last of all, it should have the capacity to represent the information that will be needed by the linguistic processing to which POS tagging is a preliminary processing phase. We give below a short description of the 36 tags of the Penn Treebank tagset (Marcus et al., 1993).

```
 1. CC  Coordinating conjunction
 2. CD  Cardinal number
 3. DT  Determiner
 4. EX  Existential there
 5. FW  Foreign word
 6. IN  Preposition or subordinating conjunction
 7. JJ  Adjective
 8. JJR Adjective, comparative
 9. JJS Adjective, superlative
10. LS  List item marker
11. MD  Modal
12. NN  Noun, singular or mass
13. NNS Noun, plural
14. NP  Proper noun, singular
15. NPS Proper noun, plural
16. PDT Predeterminer
17. POS Possessive ending
18. PP  Personal pronoun
19. PP$ Possessive pronoun
20. RB  Adverb
21. RBR Adverb, comparative
```

```
22. RBS Adverb, superlative
23. RP  Particle
24. SYM Symbol
25. TO  to
26. UH  Interjection
27. VB  Verb, base form
28. VBD Verb, past tense
29. VBG Verb, gerund or present participle
30. VBN Verb, past participle
31. VBP Verb, non-third person singular present
32. VBZ Verb, third person singular present
33. WDT Wh-determiner
34. WP  Wh-pronoun
35. WP$ Possessive wh-pronoun
36. WRB Wh-adverb
```

The selection of the linguistic features from the lexical descriptions and how they are associated to POS tags is always a difficult choice. Arbitrary linguistic choices, the application for which tagging is done, the performance expected of the tagger, and finally the disambiguation power offered by the current language technology are all important factors in determining lexical feature selection. For instance, Chanod and Tapanainen (1995) have shown that one way to improve the performance of a POS tagger for French, is to exclude the gender information from the tags of nouns and adjectives (there is less ambiguity to solve, and therefore less chance for the tagger to make an error). The gender information can be recovered afterwards by means of a lexicon and a few rules (Tufis, 1999).

It is very difficult to draw a precise boundary around the morphosyntactic information associated with POS tags, since it concerns morphology (e.g., verb tense), morphosyntax (e.g., noun/verb distinction), syntax (e.g., identification of the case for pronouns, accusative versus dative), and semantics (e.g., distinction between common and proper noun). Often it is represented by *lexical descriptions* which make explicit the way linguistic features are organised into a hierarchy and the constraints that exist between them (some features are defined only for some specific morphosyntactic categories, like the notion of tense which is restricted to the category of verbs). Here is an example of a lexical description of the word form "results":

```
[ word form = ``results''
       [ category = noun
                    subcategory = common
            morphology = [ number = plural
                           gender = neuter[2]
                                    lemma = ``result'' ]]
```

```
[ category = verb
          subcategory = main
     morphology = [ form = indicative
                    tense = present
                    number = singular
                    person = third
                            lemma = ``result'']]]
```

POS tagging is said to be one of the easiest linguistic tasks to implement, since the performance level that one can get with simple algorithms is several orders of magnitude above human performance in terms of speed and very near the level of human performance in terms of quality. Most of the complex linguistic phenomena that lie beyond the range of the current language technology occur relatively rarely. In fact, the apparent high performance level displayed by taggers in general is slightly misleading, since it is the result of the preponderant number of unambiguous word forms over the ambiguous ones in natural language. For instance, when we look at the performance on a per tag basis of one the best systems in the GRACE (Adda et al., 1999) evaluation campaign of French POS taggers, the error rate is 0.03% (4 tagging errors over 13,246 occurrences) for the punctuation category, while it goes up to 7% (1,449 tagging errors over 20,491 occurrences) for the noun category. Charniak et al. (1993) showed that the simple strategy of selecting the most likely tag for each word correctly tagged 90% of the word forms present in its data. The difficulty of POS tagging also varies greatly with the language considered; for instance, the fact that nouns are capitalized in German texts helps a lot. But problems arise from the morphological productivity of German, which results in a large number of lexical parameters, at least in the standard Markov model approach (Schmid, 1995). How to measure the performance of POS taggers is precisely the topic addressed in Section 4.

## 2 Parsing

Parsing is an analysis task aiming at identifying any constraint that controls the arrangement of the various linguistic units into sentences, and hence the ordering of words. An automatic parser tries to extract from the textual data it is given as input a description of the organization and function of the linguistic elements it finds in the data. The syntactic description can then be used by the application for which the parser was developed.

In Natural Language Processing (NLP), parsing has been studied since the early 1960s, first to develop theoretical models of human language syntax and general "deep"[3] parsers. After a period during which the formalisms have evolved to take into account more and more lexical information (linguistic descriptions anchored in words), the last decade has seen a regain of interest

in "shallow parsers" since for many applications deep configurational analyses of a sentence are completely irrelevant. Often shallow parsers are qualified in the literature as "robust". But one should not think that robustness is implied by a shallow analysis. It is true that since the function of a shallow parser is not to produce a full analysis, the number of constraints it must satisfy ought to be less than for a deep parser. Consequently, its chances of producing a valid analysis ought to be better. However, for any system this reasoning remains a hypothesis until proper tests have been conducted to assess the robustness of the parser considered. In parallel, the past few years have seen the emergence of the concept of a "treebank", a large corpus, fully annotated with deep syntactic information (see Cieri, Chapter 8, this volume), and of great value for machine learning and evaluation.

Today parsing a sentence can be approached from two different directions: first, there are the *constituent*-based models, which put the emphasis on categorial aspects of the linguistic units; second, there are the *dependency*-based models, for which the elements of interest are the syntactic functions of the linguistic units.

With constituent-based analysis, the structure of a sentence is represented by nested constituents, tagged with their syntactic category (noun phrase, verb phrase, etc.) In this model, the syntactic functions are derived from the relations existing in the constituents structure. For each syntactic function there is a particular constituent configuration: for instance, the derivation of a noun phrase (NP) from a sentence constituent indicates that the NP has the subject function. Here is an example of constituent annotation from Monceaux, (2002) (translation: *Jean looks like Paul*):

[S [NP Jean] [VP [V ressemble] [PP [Prep à] [NP Paul]]]]

In the dependency model introduced by Tesnière (1966), structural connections between the words fall into two classes: dependency relations (subordination) and junction relations (coordination). A dependency relationship is established between two words or linguistic units as soon as the syntactic and semantic features of one word constrain the possibilities for the other to co-occur. In this model, syntactic analysis is performed from left to right, and syntactic functions are carried out by specific words, i.e., the heads, and not by the constituents, as is the case with the constituent-based model. Figure 1 shows an example of dependency annotation.

Constituent models and dependency models are considered globally complementary since they offer two different points of view on the same data, and equivalent since it is theoretically possible to perform an automatic conversion (Bohnet and Seniv, 2004) in both directions, but sometimes this conversion is quite complex. We will now briefly present a few syntactic formalisms among the ones encountered most frequently in the literature, without in any way trying to be exhaustive. Ait-Mokhtar and Chanod (1997)

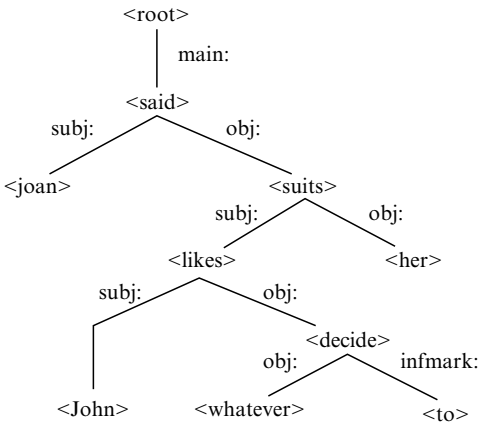


*Figure 1.* An example of dependency annotation of the sentence "John likes to decide whatever suits her" from Monceaux (2002).

describe a parser realised with finite state automata. An introduction to the use of statistical methods for parsing is proposed in Manning and Schütze (2002). A presentation of the various approaches that have been tried for parsing along with the main milestones of the domain is given in Wehrli (1997) and Abeillé and Blache (2000); in Abeillé (1993) we find a description of all the formalisms that were inspired from logic programming (based on unification operation) like the "lexical functional grammar" (LFG), the "generalized phrase structure grammar" (GPSG), the "head-driven phrase structure grammar" (HPSG), and the "tree adjoining grammar" (TAG).

LFG is a lexical theory that represents grammatical structure by means of two kinds of objects linked together by correspondences: the functional structures (f-structures), which express grammatical relations by means of attribute-value pairs (attributes may be features such as tense, or functions such as subject); and the constituent structures (c-structures), which have the form of phrase structure trees. Information about the c-structure category of each word as well as its f-structure is stored in the lexicon. The grammar rules encode constraints between the f-structure of any non-terminal node and the f-structures of its daughter nodes. The functional structure must validate the completeness and coherence condition: all grammatical functions required by a predicate must be present but no other grammatical function may be present.

In GPSG, phrase structure is encoded by means of context-free rules, which are divided into immediate dominance rules and linear precedence rules. The formalism is equipped with the so-called slash feature to handle unbounded movements in a context-free fashion. GPSG offers a high level, compact representation of language at the cost of sometimes problematic computation.

HPSG is a lexical formalism, in which language is a system of types of linguistic objects (word, phrase, clause, etc.) rather than a collection of sentences. HPSG represents grammar by declarative constraints. A grammar is a multiple-inheritance hierarchy of linguistic types. The lexicon is a subhierarchy in the grammar. A linguistic object type is represented by an underspecified feature structure, while a fully specified one identifies a unique linguistic object of the type considered. Constraints are resolved by feature structure unification.

TAG is a lightly context-sensitive formalism that represents grammar with two types of syntactic trees: elementary and auxiliary. Elementary trees hold the lexical information. In each elementary tree, a unique lexical item is attached to a leaf node. Auxiliary trees encode constraints on phrase structures. Trees are combined by means of two operations: substitution (replacement of a leaf node by a tree) and adjunction (replacement of a non-leaf node by a tree). Substitution and adjunction are constrained both by the labels of the nodes involved and by unification of the linguistic information stored in the feature structures associated to nodes.

A state-of-the-art description of dependency grammar is presented in Kahane (2000). Blache (2001) explores the contribution of contraint solving to parsing. Clément (2003) presents the latest development in parsing research. Vergne (2002) presents a multilingual parser that uses very few linguistic resources.

Parsing is an attempt at linking the linguistic phenomena naturally occurring in corpora with their encoding in a given syntactic formalism. We will see in Section 5 how evaluation attempts to qualify the way such linking is done.

## 3 Evaluation and Natural Language Processing

The purpose of evaluation is to provide an assessment of the value of a solution to a given problem; in our case, the purpose is to determine the performance of the POS tagging function or of the parsing function used in an application. When evaluating, we need to identify precisely the subject of evaluation. However, for NLP it is rather difficult to identify in a complete system, independent variables representative of the function to be observed. Often in NLP systems, the various functions involved are tightly coupled. When evaluating, the need to take into account the operational set-up adds an extra factor of complexity. This is why Sparck Jones and Galliers (1995), in their analysis and review of NLP system evaluation, stress the importance of distinguishing evaluation criteria relating to the language-processing goal (*intrinsic* criteria) from the ones relating to its role with respects to the purpose of the whole set-up (*extrinsic* criteria). One of the key questions is whether the operational set-up requires the help of a human, in which case, evaluation will also have to take into account human variability in the test conditions. The European project

EAGLES (King and Maegaard, 1998) used the role of the human operator as a guide to recast the question of evaluation in terms of users' perspective. The resulting evaluation methodology is centred on the consumer report paradigm. EAGLES distinguishes three kinds of evaluation:

1. *Progress evaluation*, where the current state of a system is assessed against a desired target state
2. *Adequacy evaluation*, where the adequacy of a system for some intended use is assessed
3. *Diagnostic evaluation*, where the assessment of the system is used to find where it fails and why

Among the other general characterisation of evaluation encountered in the literature, we retained the following ones, useful for comparing evaluation methodologies:

1. *Blackbox* or *whitebox* evaluation – whether only the global function performed between the input and output of a systems is accessible, or whether all its subfunctions are also accessible for investigation
2. *Subjective* or *objective* evaluation – whether the measurement is performed directly on data produced by the process under test, or whether it is based on the perception that human beings have of the process under test
3. *Qualitative* or *quantitative* evaluation – whether the result is a label descriptive of the behaviour of a system or whether it is the value resulting from the measurement of a particular variable
4. *Technology* or *user-oriented* evaluation (see King, Chapter 5, this volume) – whether one measures the performance of a system on a generic task (the specific aspects of any application, environment, culture, and language being abstracted as much as possible from the task), or whether one considers the actual performance of a system in the framework of a specific application, environment, culture, and language, in which case, not only technical aspects are compared, but also usability criteria like the human/machine synergy

An important point is whether the performance of a language-processing system is measured against a theoretical objective (the maximal performance value defined by the evaluation metrics), or rather against the performance level displayed by a human performing the task under consideration, as Peak (2001) proposes to do when evaluating spoken language dialogue systems.

Since the goal of evaluation is to provide answers to questions raised about the working of a given information-processing system, it is very likely that some decisive questions may have nothing to do with the ability to process a particular language. They may concern issues like software portability (choice of programming language, operating system compatibility, interoperability with other modules), or the capability of the system to handle various languages. On one occasion, decision makers preferred to select a unique multilingual system over a set of monolingual systems, for maintainability reasons, even though the multilingual system displayed lower performance on some language than its specific counterpart.

Finally we can say that any evaluation dealing with language processing resolves itself to proposing (partial) answers to the following three questions:

1. Which linguistic phenomena need to be taken into account and how frequently do they occur?
2. What kind of analysis is performed on them?
3. How will the result of their analysis be used by the application considered?

Note that, in practice, the question of which linguistic phenomena to adopt not only concerns the phenomena subject to the language processing considered, but also deals with the definition of more basic elements like affixes, word stems, types, lemmas, syntactic chunks, phrases, sentences, paragraphs, or even documents. Unfortunately, no standard exists for these.

Very often the evaluation process is based on a corpus[4] (Kilgarriff and Grefenstette, 2003). Thus we can have reproducible tests, if no human intervention is required by the application under test. If the latter cannot be achieved, a solution is to record the human intervention and reuse it at a later time. Thus the working of the application can be reproduced exactly. Fortunately, there is now enough knowledge available from corpus linguistics to ensure that a given corpus is representative of the language phenomena corresponding to the evaluation task.

The aim of this chapter is to introduce the reader to the evaluation of POS taggers and parsers for natural language textual data.[5] We will look at both POS tagging and parsing, two kinds of analysis almost always brought into play when processing natural language data.

With the current state of NLP technology, POS tagging and parsing deal essentially with the appearance of words, relegating semantic and pragmatic issues to other processing stages. Note that POS taggers and parsers are among the more readily available kinds of NLP software.

More precisely, by POS tagging is usually meant the identification of the morphosyntactic class of each word form[6] using lexical and contextual

information. The classes are either a refinement of the ones inherited from the Latin grammar (where, for instance, the class of nouns regroups the words designating entities, objects, notions, and concepts), inferred from statistical data according to an arbitrary feature set, or a mix of both of the previous cases.

By definition, the task of parsing aims at identifying any constraint that controls the arrangement of the various linguistic units into sentences, and hence the ordering of words.

If we use basic linguistic terminology in the example of "The program prints results", POS tagging will identify the word form "prints" as a verb, at the third person singular of the indicative present tense (and not as a noun), and parsing will tell that the form "program" is the subject of the verb form "prints", and that the form "results" is the direct object complement of the verb form "prints".

Note that the majority of parsing algorithms require the result of a preliminary POS tagging analysis or incorporate a POS tagging function. Note also, that the definitions we have just given of POS tagging and parsing rely on the definition of what constitutes a word, a not so trivial task as we will see in Section 3.1.

## 3.1 Identifying the Boundaries of Basic Linguistic Units

"What is a word?" (Grefenstette and Tapanainen, 1994) is a trivial question, it seems, but we will see that it is not the case. Usually, the transformation of a character stream into the sequence of basic units that any analysis requires is called *tokenisation*, and the basic units *tokens*. They are built on the basis of purely orthographic considerations, taking into account exclusive character classes, namely separators versus non-separators (Habert et al., 1998; Manning and Schütze, 2002). However, no one-to-one correspondence exists between the tokens and the word forms (Adda et al., 1997). Despite the help provided by separator characters (for the languages whose written form has them[7]), the correct identification of the various word forms cannot be done only on the basis of their appearance because language is ambiguous by nature. To perform word segmentation, the use of syntactic or semantic and sometimes even pragmatic knowledge may be required. Such knowledge is generally not available during tokenisation, since it implies the prior identification of the various word forms present. For instance, in the following examples recourse to syntax and semantics is required to distinguish between the two analyses of"of course", a noun preceded by a preposition in the first excerpt, and an adverb in the second one:

1. Early parental absence as an indicator [of] [course] and outcome in chronic schizophrenia.
2. This is an impossibility [of course] and the manufacturers admit so in private.

Since evaluation generally implies comparing several systems or different versions of the same system, it is very likely that each will use its own specific word segmentation. Segmentation variation could have an impact either on the POS tagging or parsing process (i.e., different segmentations produce different analyses), or on the performance measurement (i.e., different word segmentations entail different performance measures). Providing different specific reference data for each system to limit the influence of word segmentation would be too costly in addition to raising questions about the universality of the evaluation results. Nevertheless, to limit the influence of word segmentation, it is possible either to take an average performance measurement across all the possible segmentations, or to choose arbitrarily a reference word segmentation, but if so, which one? Alternatively, the various word segmentations can be mapped onto a common underlying token segmentation that serves as the reference segmentation. Adda et al. (1999) propose to represent explicitly the word segmentation information through indices associated to the tokens,[8] which Cloeren (1999) calls *ditto* tags. With this scheme, any word segmentation can be represented, provided that the smallest word of any segmentation has a size equal to, or larger than, the one of the smallest token.

However, using the token segmentation instead of the word segmentation for counting correct/incorrect events distorts the counts. For instance, with such a scheme an erroneous word made of two tokens will be counted twice instead of once (see Table 1). In general, the distortion introduced by the change of segmentation is somehow compensated by the fact that it applies to both the erroneous cases and the correct ones. Thus, even though the values of the event counts are different for each of the two segmentations, the relative positions of the various systems in the performance graph are often preserved across segmentation change.

The problem of splicing the input stream is not limited to small linguistic units like word forms, but concerns also larger units like sentences. Neither

*Table 1.* Example of error amplification when using token segmentation instead of word segmentation (2 errors instead of one).

| | | |
|---|---|---|
| System output | [of course] adjective | 1 error |
| Normalised system output | [of] adjective/1.2 [course] adjective/2.2 | 2 errors |
| Reference | [of] adverb/1.2 [course] adverb/2.2 | – |

a standard nor a clearly established definition of what constitutes a sentence exists. Furthermore, sentence segmentation may or may not be part of the function performed by a parser. For instance, Brill's tagger (Brill, 1995) expects to receive input that is already segmented into sentences. The quality of the sentence segmentation has a direct bearing on the quality of the parsing, since the beginning and end of sentences are elements of context that strongly determine parsing. If sentence segmentation is considered solved by some (Mikheev, 2000), this holds only for written language of good quality where punctuation marks obey typographic rules most of the time. It is quite another thing for emails or speech transcriptions. For instance, in the EASY-EVALDA evaluation campaign for parsers of French of the TECHNOLANGUE program (Mapelli et al., 2004), the sentence boundaries for the manual speech transcription[9] data had to be set by hand only after the reference syntactic annotation had been done, since the annotators needed the syntactic information to assign end-of-sentence markers in a consistent manner.

Sometimes, it may even be the document boundary which is problematic, for instance when segmenting a continuous audio stream (Gauvain et al., 2001), where the limits of the different programmes (news, advertising, shows, etc.) need to be identified.

## 4 POS Tagging Evaluation Methodology

Accuracy is certainly the most intuitive and the most used among the performance measures mentioned in the literature. It is defined as the ratio of the number of word forms correctly tagged over the total number of word forms tagged.[10] Note that the exact signification of this measure depends on what is meant exactly by "correct" tagging, the simplest definition of which requires that the following two conditions be met:

1. The word segmentation convention used by the tagger must be the same as the one used for the reference data, otherwise there is a need to deploy realignment procedures (cf. Adda et al., 1999).

2. The tagset used by the tagger must be the same as the one used to annotate the reference data, otherwise specific mapping procedures need to be applied (cf. Adda et al., 1999).

For POS tagging, everybody agrees that the *accuracy* of a tagger cannot be properly evaluated without a comparison with an annotated reference corpus, which has a distribution of linguistic phenomena that is representative of the POS tagger target application. A test suite can give interesting insights on the way the tagger handles particular linguistic phenomena. However, the relatively small size of the test suites (up to a few thousand words in general), compared to the one of a corpus (at least a million words; Paroubek and

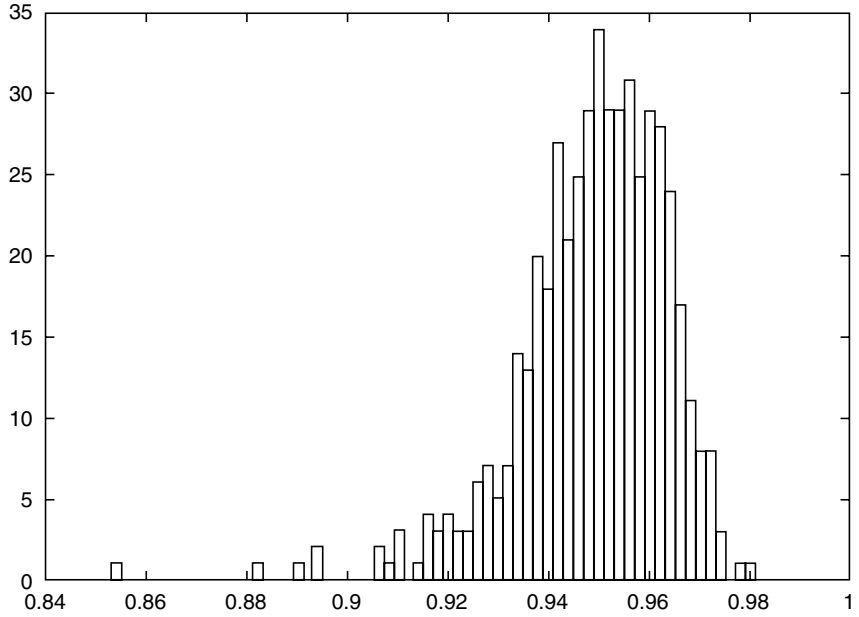


*Figure 2.* Variation of POS tagging accuracy depending on text genre. The graph (Illouz, 2000) gives the number of texts of a given genre (ordinate) in function of tagging precision (abscissa), measured on the Brown corpus (500 texts of 2000 words), with the Tree Tagger using the Penn Treebank tagset.

Rajman, 2002) does not permit to obtain enough information either on the language coverage or on the robustness of the tagger.

Not only the size of the corpus, but also its type can have an influence on the accuracy measure. To show how the performance of a POS tagger varies depending on the kind of data it processes, we give in Figure 2 the variation of tagging accuracy of the Tree Tagger (a freely available probabilistic POS tagger which uses the Penn Treebank tagset) as a function of the text genre, measured on the Brown corpus (500 texts of 2,000 words each). The accuracy varies from 85% to 98% with an average value of 94.6% (Illouz, 2000). Of course, it is recommended for testing to use material different from that which served for training of the system, since performance will invariably be better on the training material (van Halteren, 1999).

Things get more complicated as soon as we start considering cases other than the one in which both the tagger and the reference data assign only one tag per token. Then the accuracy measure no longer permits a fair comparison between different taggers, if they are allowed to propose partially disambiguated taggings. Van Halteren (1999) proposes in such cases to use

the *average tagging perplexity*, i.e., the average number of tags per word assigned by the system,[11] or to have recourse to *precision* and *recall*, the now well-known evaluation measures from Information Retrieval.

Let us denote with $t_i$ the set of tags assigned to the $i^{\text{th}}$ word form $w_i$ by a tagger and $r_i$ the set of tags assigned to the same word form in the reference annotations. The value of the precision and recall for this word form are, respectively, the ratio of the number of correct tags over the number of tags assigned by the system $P(w_i) = \frac{|t_i \cap r_i|}{|t_i|}$, and the ratio of the number of correct tags over the number of tags assigned in the reference $R(w_i) = \frac{|t_i \cap r_i|}{|r_i|}$. By averaging the respective sums of the two previous quantities for all the word forms, we obtain the measures over the whole corpus $P = \frac{1}{N} \sum_{i=1}^{N} p_i$ and similarly for R. Often precision and recall are combined together into one single value, the *f-measure* whose formula accepts as parameter $\alpha$ the relative importance[12] given to precision over recall, $F = \frac{1}{\frac{\alpha}{P} + \frac{(1-\alpha)}{R}}$ (Manning and Schütze, 2002).

In the very frequent case where only one tag per word form is assigned in the reference annotation, precision and recall take very intuitive interpretations. Recall is the proportion of word taggings holding one correct tag. Precision is the ratio between the recall and the average number of tags assigned per word by the tagger. This second measure is relatively close to the *average ambiguity* (Tufis and Mason, 1998), the average number of tags assigned by a lexicon to the words of a corpus. It integrates both the a priori ambiguity of the corpus and the *delicacy*[13] of the tagset used in the lexicon. Average ambiguity can be used to quantify the relative difficulty offered by the task of tagging the corpus, i.e., how much ambiguity remains to be solved, since some word forms have already an unambiguous tagging in the lexicon.

Note that precision is a global performance measurement which does not give any information about the error distribution over the various linguistic phenomena or the various genres of text, or on the types of error. It is not because two taggers have similar precision values that they make the same errors at the same locations. Therefore, it may be of interest to quantify the similarity between two taggings of the same text. There exists a measure initially developed for this very purpose, but for human annotators. It is the $\kappa$ (kappa) coefficient (Carletta, 1996), which compensates for the cases where the two taggings agree by chance.

Other approaches use measures from Information Theory (Resnik and Yarowsky, 1997), like the per word cross-entropy, which measures the distance between a stochastic process q and a reference stochastic process p. In this approach, tagging is considered to be a stochastic process which associates to each word form a probability distribution over the set of tags. If we suppose that the reference process is stationary[14] and ergodic,[15] and that two subsequent taggings are two independent events, then for a sufficiently large corpus, the cross-entropy can be easily computed (Cover and Thomas, 1991).

Let us mention another set of measures which has been used in the GRACE evaluation campaign (Adda et al., 1999): precision and decision. The precision measures the number of times a word was assigned a single correct tag. The decision measures the ratio between the number of words which have been assigned a single tag and the total number of words. The originality of this measure lies with the possibility to plot the whole range of performance values reachable by a system, if one were to attempt to disambiguate some or all of the taggings that were left ambiguous by the tagger.

In the literature, most of the results mention precision values which are almost always greater than 90% and sometimes reach 99%. Already in de Rose (1988), the Volsunga tagger had achieved 96% precision for English on the Brown corpus. The best result in the GRACE evaluation of French taggers was 97.8% precision on a corpus of classic literature and the *Le Monde* newspaper. In the same evaluation, a lexical tagging (assigning all the tags found in the lexicon associated to the considered word form) achieved 88% precision. This result dropped to 59% precision[16] when a few contextual rule files were applied to try to artificially reduce the ambiguous taggings to one single tag per word. But let us remind the reader that all these measures must be considered with caution since they highly depend on the size and composition of the tagset as well as on the segmentation algorithms and on the genre of the text processed. Furthermore, evaluation results are given on a per word basis, which is not necessarily an appropriate unit for some applications where units like the sentence, the paragraph, or the document are often more pertinent. For instance, for a 15-word sentence and a tagging precision of 96% at the word level, we only get a tagging precision of 54.2% at the sentence level, i.e., almost 1 sentence in 2 contains a tagging error. Conversely, to achieve a 95% tagging precision at the sentence level, we would need to have a tagger which would achieve a 99.67% precision at the word level.

Although POS tagging seems to be a task far simpler than parsing, a POS tagger is a complex system combining several functions (tokeniser, word/sentence segmenter, context-free tagger, POS tag disambiguator) which may use external linguistic resources like a lexicon and a tagset. Evaluating such systems implies clear choices about the criteria that will be effectively taken into account during evaluation. Evaluation cannot resume itself to the simple measurement of tagging accuracy; factors like the processing speed (number of words tagged per second), the software portability (on which operating system can the tagger run, how easily can it be integrated with other modules), its robustness (is the system tolerant to large variations of the input data characteristics), the delicacy of the tagset (how fine a linguistic distinction can be made between two word classes), and the multilingualism of the system all constitute different dimensions of the evaluation space, the importance of which varies depending on the purpose of evaluation.

## 5 Methodology and Evaluation Measures for Parsing

Historically, the first comparative evaluation of the output of automatic parsers has been done by human experts, who formulated a diagnostics based on the processing of a set of test sentences. Very often this way of performing evaluation implies the use of an analysis grid (Blache and Morin, 2003) that lists evaluation features. To our knowledge the first publication on the subject for French is from Abbeillé (1991). In order to limit the bias introduced by the views of a particular expert and to promote reuse of linguistic knowledge, the community started to devise test suites compare, for instance, the European project TSNLP (Oepen et al., 1996). It produced a syntactic test suite for several European languages, with each test suite containing both positive and negative examples of annotations, classified by linguistic phenomena. Although they are of a great help to experts, test suites have nevertheless several drawbacks. First, they do not reflect the statistical distribution of the phenomena encountered in real corpora and they are also too small to be reused for evaluation (except for non-regression tests), because once they have been disclosed, it is relatively easy to customise any parser for the specific examples contained in the test suite. The second drawback concerns the formalism, because very likely the test suite and the parser under test will use different syntactic formalisms; thus a mapping between the formalisms will be required, which may generate some information loss. To answer this criticism, a new approach inspired by statistics and machine learning has emerged, helped by the recent progress in NLP and the development of standards for mark-up, i.e., the treebanks. A treebank is a relatively large corpus (at least more than 1 million word forms), completely annotated with a particular formalism in a consistent way. The first and certainly the most famous is the Penn Treebank (Marcus et al., 1993), which has inspired other developments like Brant et al. (2002) and Abbillé et al. (2000) for French. However, while treebanks provide a solution to the problem of language coverage, they do not solve the main problem of parsing evaluation, i.e., which pivot formalism should we use to obtain a faithful evaluation? A faithful evaluation is an evaluation that preserves both the information present in the reference data and in the data output by the parser. It should also provide the means to describe all the linguistic phenomena of the test data. Defining such a formalism is precisely one of the objectives of parsing, i.e., providing a universal formalism for all the phenomena of a language.

Up to now many propositions have been made in that direction. Some use annotation mappings (Gaizauskas et al., 1998); others propose to compare information quantity (Musillo and Simaán, 2002), which unfortunately obliges one to build a parallel corpus per formalism; and others again propose to use

automatic grammar-learning procedures (Xia and Palmer, 2000) or computations based on the "edit" distance (Roark, 2002). The oldest approach (Black et al., 1991) focused on evaluation measures and used the constituent boundaries to compare parsers by measuring the percentage of crossing brackets (number of constituent boundaries output by the parser that cross[17] a constituent boundary of the reference) and recall (number of constituent boundaries output by the parser that exist in the reference data). Precision was added to the two previous measures to constitute what was called the GEIG[18] scheme (Srinivas et al., 1996) or PARSEVAL measures (Carroll et al., 2002). Unfortunately these measures were applicable in practice only on unlabelled constituents, i.e., without any information as to which category the constituent belongs to, since the output of the parsers that participated in these experiments was too diverse to allow for the use of such information. The PARSEVAL scheme takes into account only part of the information produced by a parser. Furthermore, it is more easily applied to constituent-based parsers.

To try to solve this problem, Lin (1998) suggested to use dependencies rather than constituents for evaluation. Briscoe et al. (2002) and Caroll et al. (1998, 2003) propose to go even further by annotating tagged grammatical relations between lemmatised lexical heads, in order to work on both the logic and grammatical relations that are present in the sentence, instead of looking at the topological details of a parse tree. The most recent developments in large-scale evaluation effort concern French with the TECHNOLANGUE program (Mapelli et al., 2004) and its evaluation campaign for parsers, EASY (Vilnat et al., 2003) of the EVALDA project, which proposes to use an annotation formalism inspired by Carroll et al. (2003) with an initial level of constituents and grammatical relations, but without any explicit notion of head (Gendner et al., 2003; Vilnat et al., 2003).

The EASY annotation scheme recognises 6 types of syntactic chunks and 14 functional relations. The xml-like tags (cf.) Figure 3 indicate syntactic

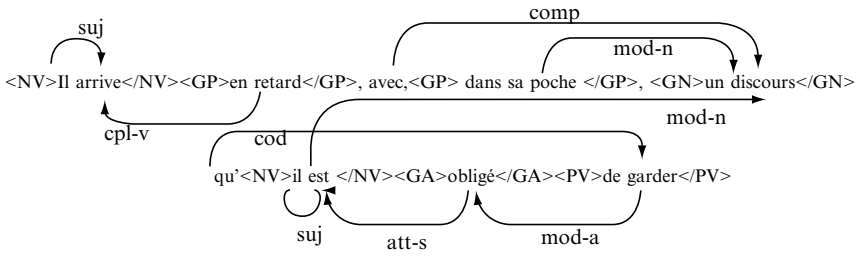


*Figure 3.* Example of reference annotation of the EASY evaluation campaign for the sentence: "He arrives late, with in his pocket, a discourse which he must keep."

chunks: $NV$ = verb chunk, including clitics, as in "Il arrive" "He comes"; $GP$ = prepositional chunk; $GN$ = nominal chunk; $GA$ = adjectival chunk; $PV$ = prepositional-verbal chunk (i.e., for infinitive forms introduced by a preposition). The arrows indicate the functional relations, relating either syntactic chunks or tokens; *suj* means subject; *comp* represents "complementiser" mainly for conjunctive subordinates, with the subordinate conjunction and the verbal chunk of the subordinate as arguments, but it is also used, like here, to annotate the relation between a preposition and a nominal chunk or verbal chunk when they cannot be annotated as $GP$ or $PV$, for instance, in the presence of an insertion ("dans sa poche", "in his pocket"); *cpl-v* means verb complement; *cod* encodes direct object; *mod-n* stands for noun modifier; *mod-a* for adjective modifier; *atb-s* means subject attribute.

## 5.1 Which Performance can Current Parsing Technology Achieve

Because there are many more different formalisms and these vary more than the ones used for POS tagging, the amount of reliable, widely available parsing software is smaller than for taggers. Even more so, since the analysis to perform is much more complex in the case of parsing. With the same reservation as for what was said about POS tagging, we will now give some results indicative of the level of performance achievable by current parsing technologies, without any claim of being exhaustive.

Black (1993) gives for five systems a percentage of correct sentences (without any constituent boundary crossing) varying from 29% to 78%. He gives for two systems respective values of 38% and 51% of exact match between the constituent boundaries of the parsers and the ones of the reference annotation. Similarly, John Carroll (Carroll et al., 2002) mentions, that describes a comparative evaluation done by the GEIG in 1992, recall measures on constituent boundaries varying from 45% to 64%, a mean rate of constituent boundary crossing between 3.17 and 1.84, and a sentence percentage, for which the best analysis contains at least one constituent boundary crossing, between 84% and 34%.

Srinivas et al. (1998) report that the XTAG (Doran et al., 1994) analyses correctly 61.4% of the sentences of the TSNLP test suite (Oepen et al., 1996) and 89.6% of the sentences of a weather forecast bulletin corpus. Srinivas et al. (1998) have achieved a precision value of 84.2% for another version of the same parser, measured on the dependencies extracted from the Penn Treebank, and Xia and Palmer (2000) computed on the same corpus a value of 97.2% of structure similarity for the syntactic patterns. Crouch et al. (2002) present values of f-measure lying between 73% and 79%, measured on the Penn Treebank for an LFG parser enhanced with a stochastic disambiguating mechanism.

*Table 2.* Performance range of four parsers of French and their combination, on questions of the Question and Answer TREC track corpus.

| | Precision | Recall |
|---|---|---|
| Noun phrase | from 31.5% to 86.6% | from 38.7% to 86.6% |
| Verb phrase | from 85.6% to 98.6% | from 80.5% to 98.6% |
| Prepositional phrase | from 60.5% to 100% | from 60.5% to 100% |

For a Category Combinatory Grammar (CCG), Clark and Hochenmaier (2002) give PARSEVAL results of 85.5% precision for unlabelled constituents (and 81.6% for labelled ones) and 85.9% recall on unlabelled constituents (and 81.9% on labelled constituents).

For French, Francopoulo and Blache (2003) have obtained a value of 74% for the f-measure with a chunk parser. Monceaux (2002) and Monceaux and Vilnat (2003) have studied the combination of parsers for the analysis of questions. The performance ranges of the four parsers and their combination are presented in Table 2.

As we have just seen, getting a clear idea of the level of performance achievable by the current parsing technology is rather difficult since the formalisms vary greatly and the results of evaluation display important differences, depending on the evaluation procedure applied and on the reference data used, even more so since evaluation results are scarce for languages other than English.

## 6 Conclusion

When POS taggers and parsers are integrated in an application, only quantitative blackbox methodologies are available to gauge their performance. This approach is characteristic for technology-oriented evaluation, which interests mostly integrators and developers, contrary to user-oriented evaluation, for which the interaction with the final user is a key element of the evaluation process.

Although the corpus-based automatic evaluation procedures do provide most of the information useful for assessing the performance of a POS tagger or parser, the recourse to the opinion of an expert of the domain is essential, not only to provide an interpretation of the results returned by the automatic evaluation procedures, but also to provide the knowledge needed to define the conditions under which the evaluation measures will be taken.

POS tagging evaluation methodology is now mature, and there exist enough results in the literature to be able to compare POS taggers on grounds sufficiently sound if one has the proper evaluation tools and an annotated corpus,

the cost of which is rather high, not only because of the manpower needed, but also because of the annotation quality required.

For parsing, the situation is less clear, possibly only because of the greater variety of the syntactic formalisms and of the analysis algorithms. It is very difficult to compare on a fair basis systems that use different formalisms. However, the situation begins to change with the emergence of new evaluation protocols based on grammatical relations (Carroll et al., 2003) instead of constituents, and large-scale evaluation campaigns, like the French EASY-EVALDA of the TECHNOLANGUE program for parsers of French (Vilnat et al., 2003).

## Notes

1. A Treebank is a large corpus completely annotated with syntactic information (trees) in a consistent way.

2. In English, gender for nouns is only useful for analysing constructions with pronouns.

3. A "deep" parser describes for all the word forms of a sentence, in a complete and consistent way, the various linguistic elements present in the sentence and the structures they form; on the contrary, a "shallow" parser only provides a partial description of the structures.

4. This is particularly true of any batch-processing activity like POS tagging and parsing.

5. Of all kinds, including emails or produced by automatic speech transcription.

6. We will refrain from using the term *type* to refer to word forms, to avoid any confusion with other meanings of this term.

7. Languages like Chinese are written without separators.

8. Tokens are indexed with indices made of the position of the current token in the compound word, associated with the total number of tokens in the compound, e.g., of/1.2 course/2.2.

9. Transcription of oral dialogues, recorded in various everyday life situations.

10. The error rate is simply the 1's complement of the accuracy.

11. Note that this measure takes all its sense when given with the corresponding measure of the standard deviation.

12. In general $\alpha = 0.5$.

13. The level of refinement in linguistic distinction offered by the tagset, in general, correlated with the number of tags: the finer the distinctions, the larger the tagset.

14. A stochastic process is stationary when its statistical characteristics do not depend on the initial conditions.

15. Observations made at any time over a succession of process states are the same as the observations made over the same states but on a large number of realisations.

16. The precision decreases, because as the ambiguous taggings are resolved, they become unambiguous and thus are taken into account in the computation of the precision, while before they were only taken into account in the measurement of the decision.

17. Here is an example where the A parentheses cross the B parentheses: $(^{\mathrm{A}}\ (^{\mathrm{B}}\ \mathrm{A})\mathrm{B})$.

18. Grammar Evaluation Interest Group.

## References

Abeillé, A. (1991). Analyseurs syntaxiques du français. *Bulletin Semestriel de l'Association pour le Traitement Automatique des Langues*, 32:107–120.

Abeillé, A. (1993). *Les nouvelles syntaxes*. Armand Colin, Paris, France.

Abeillé, A. and Blache, P. (2000). *Grammaires et analyseurs syntaxiques*, pages 61–76, Ingénierie des langues, Hermes Science Publication, Paris, France.

Abeillé, A., Clément, L., and Kinyon, A. (2000). Building a Treebank for French. In *Proceedings of the Second International Conference on Language Ressources and Evaluation (LREC)*, pages 1251–1254, Athens, Greece.

Adda, G., Adda-Decker, M., Gauvain, J.-L., and Lamel, L. (1997). Text Normalization and Speech Recognition in French. In *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech)*, volume 5, pages 2711–2714, Rhodes, Greece.

Adda, G., Mariani, J., Paroubek, P., Rajman, M., and Lecomte, J. (1999). L'action grace d'évaluation de l'assignation des parties du discours pour le français. *Langues*, 2(2):119–129.

Ait-Mokhtar, S. and Chanod, J.-P. (1997). Incremental Finite-State Parsing. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 72–79, Washington, DC, USA.

Blache, P. (2001). *Les grammaires de propriétés: des contraintes pour le traitement automatique des langues*, Hermes Science Publication, Paris, France.

Blache, P. and Morin, J.-Y. (2003). Une grille d'évaluation pour les analyseurs syntaxiques. In *Acte de l'atelier sur l'Evaluation des Analyseurs Syntaxiques dans les actes de la 10$^e$ Conférence Annuelle sur le Traitement Automatique des Langues Naturelles (TALN)*, volume II, pages 77–86, Batz-sur-Mer, France.

Black, E. (1993). Parsing English by Computer: The State of the Art. In *Proceedings of the International Symposium on Spoken Dialog*, pages 77–81, Tokyo, Japan.

Black, E., Abney, S., Flickenger, D., Gdaniec, C., Grishman, R., Harison, P., Hindle, D., Ingria, R., Jelineck, F., Klavan, J., Liberman, M., Marcus, M., Roukos, S., Santorini, B., and Strzalkowski, T. (1991). A Procedure for Quantitatively Comparing the Syntactic Coverage of English Grammars. In *Proceedings of the Fourth DARPA Speech and Natural Language Workshop*, pages 306–311, Morgan Kaufman, Pacific Grove, California, USA.

Bohnet, B. and Seniv, H. (2004). Mapping Dependency Structures to Phrase Structures and the Automatic Acquisition of Mapping Rules. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)*, pages 855–858, Lisboa, Portugal.

Brant, S., Dipper, S., Hansen, S., Lezius, W., and Smith, G. (2002). The TIGER Treebank. In *Proceedings of the First Workshop on Treebank and Linguistics Thories (TLT)*, pages 24–41, Sozopol, Bulgaria.

Brill, E. (1995). Transformation-Based Error Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging. *Computational Linguistics*, 21(4):543–565.

Briscoe, E., Carroll, J., Grayham, J., and Copestake, A. (2002). Relational Evaluation Schemes. In *Proceedings of the Workshop Beyond Parseval – Towards Improved Evaluation Measures for Parsing Systems at the Third International Conference on Language Resources and Evaluation (LREC)*, pages 4–8, ELRA, Las Palmas, Gran Canaria, Spain.

Carletta, J. (1996). Assessing Agreement on Classification Tasks: The Kappa Statistics. *Computational Linguistics*, 22(2):249–254.

Carroll, J., Briscoe, T., and Sanfilipo, A. (1998). Parser Evaluation: A Survey and a New Proposal. In *Proceedings of the First International Conference on Linguistic Resources and Evaluation (LREC)*, pages 447–454, Granada, Spain.

Carroll, J., Frank, A., Lin, D., Prescher, D., and Uszkoreit, H. (2002). Beyond Parseval – Towards Improved Evaluation Measures for Parsing Systems. In Carroll, J., editor, *Proceedings of the Workshop Beyond Parseval – Towards Improved Evaluation Measures for Parsing Systems at the Third International Conference on Language Resources and Evaluation (LREC)*, pages 1–3, ELRA, Las Palmas, Gran Canaria, Spain.

Carroll, J., Minnen, G., and Briscoe, E. (2003). *Parser Evaluation Using a Grammtical Relation Annotation Scheme*, pages 299–316, Treebanks: Building and Using Parsed Corpora, Kluwer, Dordrecht, The Netherlands.

Chanod, J.-P. and Tapanainen, P. (1995). Creating a Tagset, Lexicon and Guesser for a French Tagger. In *Proceedings of the ACL SIGDAT Workshop From Text to Tags: Issues in Multilingual Analysis*, pages 58–64, University College, Dublin, Ireland.

Charniak, E., Hendrickson, C., Jacobson, N., and Perkowitz, M. (1993). Equations for Part of Speech Tagging. In *Proceedings of the the 11$^{th}$ Conference of the American Association for Artificial Intelligence (AAAI)*, pages 784–789, Washington DC, USA.

Clark, S. and Hochenmaier, J. (2002). Evaluating a Wide-Coverage CCG Parser. In *Proceedings of the Workshop Beyond Parseval – Towards Improved Evaluation Measures for Parsing Systems at the Third International Conference on Language Resources and Evaluation (LREC)*, pages 60–66, ELRA, Las Palmas, Gran Canaria, Spain.

Clément, L. (2003). Evolution en analyse syntaxique. *Revue TAL*, 44(3). Hermes Science Publication, Paris, France.

Cloeren, J. (1999). Tagsets. In van Halteren, H., editor, *Syntactic Wordclass Tagging*, chapter 4, pages 37–54, Kluwer Academic Publishers, Dordrecht, The Netherlands.

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. John Wiley, New York, USA.

Crouch, R., Kaplan, R., King, T., and Riezler, S. (2002). Comparison of Evaluation Metrics for a Broad Coverage Stochastic Parser. In *Proceedings of the Workshop Beyond Parseval – Towards Improved Evaluation Measures for Parsing Systems at the Third International Conference on Language Resources and Evaluation (LREC)*, pages 67–74, ELRA, Las Palmas, Gran Canaria, Spain.

de Rose, S. J. (1988). Grammatical Category Disambiguation by Statistical Optimization. *Computational Linguistics*, 14(1):31–39.

Doran, C., Egedi, D., Hockey, B., Srinivas, B., and Zaidel, M. (1994). XTAG System – A Wide Coverage Grammar for English. In *Proceedings of the 17$^{th}$ International Conference on Computational Linguistics (COLING)*, pages 922–928, Kyoto, Japan.

Francopoulo, G. and Blache, P. (2003). Tag chunker, mécanisme de construction et évaluation. In *Acte de l'atelier sur l'Evaluation des Analyseurs Syntaxiques dans les actes de la 10$^{e}$ Conférence Annuelle sur le Traitement Automatique des Langues Naturelles (TALN)*, pages 95–104, Batz-sur-Mer, France.

Gaizauskas, R., Hepple, M., and Huyck, C. (1998). Modifying Existing Annotated Corpora for General Comparative Evaluation of Parsing. In *Proceedings of the Workshop on Evaluation of Parsing Systems in the Proceedings of the First International Conference on Language Resources and Evaluation (LREC)*, pages 21–28, Granada, Spain.

Gauvain, J.-L., Lamel, L., and Adda, G. (2001). Audio Partitioning and Transcription for Broadcast Data Indexation. *MTAP Journal*, 14(2):187–200.

Gendner, V., Illouz, G., Jardino, M., Monceaux, L., Paroubek, P., Robba, I., and Vilnat, A. (2003). Peas, the First Instantiation of a Comparative Framework for Evaluating Parsers of French. In *Proceedings of the Tenth Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 95–98, Budapest, Hungary. Companion Volume.

Grefenstette, G. and Tapanainen, P. (1994). What is a Word, What is a Sentence? Problems of Tokenization. In *Proceedings of the Third International Conference on Computational Lexicography*, pages 79–87, Budapest, Hungary.

Habert, B., Adda, G., Adda-Decker, M., de Mareuil, P. B., Ferrari, S., Ferret, O., Illouz, G., and Paroubek, P. (1998). The Need for Tokenization Evaluation. In *Proceedings of the First International Conference on Language Resources and Evaluation (LREC)*, volume 1, pages 427–431, Granada, Spain.

Illouz, G. (2000). Sublanguage Dependent Evaluation: Toward Predicting NLP Performances. In *Proceedings of the Second International Conference on*

*Language Ressources and Evaluation (LREC)*, pages 1251–1254, Athens, Greece.

Kahane, S. (2000). Les grammaires de dépendance. *Revue TAL*, 41(1):318. Hermes Science Publication, Paris, France.

Kilgarriff, A. and Grefenstette, G. (2003). Introduction to the Special Issue on the Web as Corpus. *Computational Linguistics*, 29(3):333–347.

King, M. and Maegaard, B. (1998). Issues in Natural Language System Evaluation. In *Proceedings of the First International Conference on Linguistic Resources and Evaluation (LREC)*, volume 1, pages 225–230, Granada, Spain.

Lin, D. (1998). Dependency-Based Evaluation of MINIPAR. In *Proceedings of the Workshop on Evaluation of Parsing Systems*, pages 33–39, Granada, Spain.

Manning, C. D. and Schütze, H. (2002). *Foundation of Statistical Natural Language Processing*. Massachusetts Institute of Technology Press, 5th edition.

Mapelli, V., Nava, M., Surcin, S., Mostefa, D., and Choukri, K. (2004). Technolangue: A Permanent Evaluation & Information Infrastructure. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)*, pages 381–384, Lisboa, Portugal.

Marcus, M., Santorini, B., and Marcinkiewciz, M. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330.

Mikheev, A. (2000). Tagging Sentence Boundaries. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 264–271, Seattle, USA.

Monceaux, L. (2002). *Adaptation d'un niveau d'analyse des interventions dans un dialogue – Application à un système de question – réponse*. Thèse de doctorat, Université Paris XI, France.

Monceaux, L. and Vilnat, A. (2003). Multi-analyse, vers une analyse syntaxique plus fiable. In *Actes de la 10e Conférence Annuelle sur le Traitement Automatique des Langues Naturelles (TALN)*, pages 215–222, Batz-sur-Mer, France.

Musillo, G. and Simaán, K. (2002). Toward Comparing Parsers from Different Linguistic Frameworks – An Information Theoretic Approach. In *Proceedings of the Workshop Beyond Parseval – Towards Improved Evaluation Measures for Parsing Systems at the Third International Conference on Language Resources and Evaluation (LREC)*, pages 44–51, Las Palmas, Gran Canaria, Spain.

Oepen, S., Netter, K., and Klein, J. (1996). Test Suites for Natural Language Processing. In Nerbonne, J., editor, *Linguistic Databases*, pages 13–36, Center for the Study of Language and Information (CSLI) Publications, Stanford, California, USA.

Paek, T. (2001). Empirical Methods for Evaluating Dialog Systems. In *Proceedings of the Workshop on Evaluation Methodologies for Language and Dialog Systems*, pages 1–8, Toulouse, France. Annual Meeting of the Association for Computational Linguistics (ACL).

Paroubek, P. and Rajman, M. (2000). Multitag une ressource linguistique produit du paradigme d'évaluation. In *Actes de la 7ème Conférence Annuelle sur le Traitement Automatique des Langues Naturelles*, pages 297–306, Lausanne, Switzerland.

Resnik, P. and Yarowsky, D. (1997). A Perspective on Word Sense Disambiguation Methods and their Evaluation. In *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What and How?*, pages 79–86, Washington, USA.

Roark, B. (2002). Evaluating Parser Accuracy Using Edit Distance. In *Proceedings of the Workshop Beyond Parseval – Towards Improved Evaluation Measures for Parsing Systems at the Third International Conference on Language Resources and Evaluation (LREC)*, pages 30–36, Las Palmas, Gran Canaria, Spain.

Schmid, H. (1995). Improvements in Part-of-Speech Tagging with an Application to German. In *Proceedings of the 14th International Conference on Computational Linguistics*, pages 172–176, Kyoto, Japan.

Sparck Jones, K. and Galliers, J. R. (1995). *Evaluating Natural Language Processing Systems*. Springer Verlag, Heidelberg, Germany.

Srinivas, B., Doran, C., Hockey, B. A., and Joshi, K. (1996). An Approach to Robust Partial Parsing and Evaluation Metrics. In Carroll, J., editor, *Proceedings of the Workshop on Robust Parsing*, pages 70–82, ESSLI, Prague, Czech Republic.

Srinivas, B., Sarkar, A., Doran, C., and Hockey, B. (1998). Grammar and Parser Evaluation in the XTAG Project. In *Proceedings of the Workshop on Evaluation of Parsing Systems*, pages 63–69, Granada, Spain.

Tesnière, L. (1966). *Élements de syntaxe structurale*, Klincksieck, Paris, France.

Tufis, D. (1999). Text Speech and Dialog. In *Tiered Tagging and Combined Classifier*, *Lecture Notes in Artificial Intelligence*, volume 1692, pages 28–33. Springer.

Tufis, D. and Mason, O. (1998). Tagging Romanian Texts: A Case Study for QTAG, a Language Independent Probabilistic Tagger. In *Proceedings of the First International Conference on Language Resources and Evaluation*, pages 589–596, Granada, Spain.

van Halteren, H. (1999). Performance of Taggers. In *Syntactic Wordclass Tagging*, pages 81–94, Kluwer Academic Publishers, Dordrecht, The Netherlands.

Vergne, J. (2002). Une méthode pour l'analyse descendante et calculatoire de corpus multilingues – application au calcul des relations sujet-verbe. In *Actes de la 9e Conférence Annuelle sur le Traitement Automatique des Langues Naturelles (TALN)*, pages 63–74, Nancy, France.

Vilnat, A., Paroubek, P., Monceaux, L., Robba, I., Gendner, V., Illouz, G., and Jardino, M. (2003). Easy or How Difficult Can It Be to Define a Reference Treebank for French. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT)*, pages 229–232, Växjö, Sweden.

Wehrli, E. (1997). *L'analyse syntaxique des langues naturelles: problèmes et méthodes*. Masson, Paris, France.

Xia, F. and Palmer, M. (2000). Evaluating the Coverage of LTAGS on Annotated Corpora. In *Proceedings of the Workshop on Using Evaluation within HLT Programs: Results and Trends, at the Second International Conference on Language Ressources and Evaluation (LREC)*, pages 1–6, Athens, Greece.