Label Pre-annotation for Building Non-projective Dependency Treebanks for French

Ophélie Lacroix, Denis Béchet, and Florian Boudin

LINA - Université de Nantes
2 Rue de la Houssinière
44322 Nantes Cedex 3
{ophelie.lacroix, denis.bechet, florian.boudin}@univ-nantes.fr

Abstract. The current interest in accurate dependency parsing make it necessary to build dependency treebanks for French containing both projective and non-projective dependencies. In order to alleviate the work of the annotator, we propose to automatically pre-annotate the sentences with the labels of the dependencies ending on the words. The selection of the dependency labels reduces the ambiguity of the parsing. We show that a maximum entropy Markov model method reaches the label accuracy score of a standard dependency parser (MaltParser). Moreover, this method allows to find more than one label per word, i.e. the more probable ones, in order to improve the recall score. It improves the quality of the parsing step of the annotation process. Therefore, the inclusion of the method in the process of annotation makes the work quicker and more natural to annotators.

1 Introduction

Dependency-based methods for syntactic parsing have become increasingly popular in natural language processing in recent years [1]. Most proposed approaches for dependency parsing are data-driven and require large sets of manually annotated sentences, called treebanks. Needless to say, annotating such data is very costly and time consuming. One usual way to alleviate the burden of manual annotation is to automatically pre-annotate the data, so that annotators only have to validate pre-annotated sentences.

Available treebanks for French are constituency treebanks that were converted into dependency ones, e.g. [2]. While the conversion method that was used is able to generate non-projective dependency structures [3], constituency trees are always projective. It is then not surprising that converted dependency trees do not reflect non-projective relations. As a consequence, data-driven dependency parsers trained on these converted treebanks fail to produce non-projective dependency structures. Developing a French treebank that contains non-projective trees is therefore necessary for improving parsing accuracy.

A relevant work in this direction is that of Dikovsky [4] who proposed a framework for jointly constructing a treebank and a grammar for French (CDGFr).

The result of this work is a treebank consisting of 3030 sentences annotated with dependency structures (projective and non-projective) along with an annotation environment called CDG Lab [5].

In CDG Lab, annotating a sentence is a three-step process. The first step is the manual pre-annotation of the sentence. It consists of selecting either a grammatical class or a dependency label for each word through a selection form. The computational time of the second step, the dependency analysis, is exponentially proportional to the number of selected labels per word. So, the selection of one label per word restrains the search space of the grammar-based analysis and then make the analysis practical. The last step is a manual validation.

Filling the selection form is a tedious task for the annotators. In this paper, we propose to automatize the sentence pre-annotation step in order to alleviate the work of the annotator through the building of large dependency treebanks. We replace the selection form by a method using a maximum entropy Markov model to provide dependency labels and selecting one or more dependency labels for each word depending on their probability score. The method reaches the label accuracy scores of a standard data-driven parser, MaltParser [6], in addition to providing more than one label per word. Moreover, this number can be controlled to impact positively the grammar-based dependency parsing. Then, the parsing step becomes a trade-off between the preservation of a high recall score and acceptable parsing time in order to reduce the error correction rate and therefore the whole time of the annotation process. Finally, the use of the automatic label pre-annotation tool facilitates and speeds up the creation of new large French dependency treebanks containing both projective and non-projective trees.

The rest of this paper is organized as follows. In Section 2, we first review the related work on methods for building dependency treebanks. In Section 3, we present the background of dependency parsing and describe the process of annotating a sentence in CDGLab. Then, we detail our automatic pre-annotation method in Section 4 and examine the results in Section 5. Finally, we discuss the benefit of the pre-annotation process in the building of dependency treebanks in Section 6 and conclude.

2 Related Work

Dependency treebanks are now available for many languages [7]. Depending on the available tools and resources, sentences may be fully or partially annotated with Part-Of-Speech tags and dependency relations.

On the one hand, conversion methods can be applied to convert constituency treebanks to dependency ones. The converted treebanks require no or very few corrections after conversion due to the quality and quantity of the syntactic and grammatical information given by the original constituency treebanks. Such kind of method has been applied to French and also to English through the building of the Penn Treebank [8].

On the other hand, the development of large treebanks requires several automatic and manual steps. The automated steps occur on various levels of analysis (segmentation, POS-tagging, parsing) and require the validation of the annotators. The benefit over the conversion methods is to be independent from other formalisms like the constituent one. For example, the annotation process of the Prague Dependency Treebank [9] includes its own level of analysis (e.g. morphological, analytical, tectogrammatical). Furthermore, many tasks on treebanks building exploit the performance of a data-driven dependency parser, such as the MaltParser [6] : this is the case for various work (e.g. for Indonesian [10], Latin [11], Turkish [12]) to pre-annotate their data.

An example of a dependency treebank built from scratch is the speech dependency treebank for French. Here, Cerisara and al. [13] perform a manual segmentation step before the tagging and parsing steps. Nevertheless, in our work, we do not want to use a converted treebank to train a model because it does not include non-projective trees. And, in order to provide trees consistent with the CDGFr, we do not use a data-driven parser.

The automatic pre-annotation process often includes POS-tagging. In the case of non-projective dependency parsing, Alfared and al. [14] showed that the upstream disambiguation of POS-tagging is not sufficient to disambiguate the grammar-based parsing. Our annotation process uses a pre-annotation step which selects one or more dependency label for each word as well as the POS. The spirit of this particular task is in the way of supertagging [15]. But, here we want to predict a single information, the dependency label rather than a complex structure like a type (Categorial Grammar) or an elementary tree (Tree-Adjoining Grammar). The complexity of this task is halfway between POS-tagging and supertagging.

3 Annotation Framework

3.1 Background

The dependency representation allows to represent both projective and non-projective relations existing in natural languages. A dependency tree containing at least one non-projective dependency is called non-projective. For a dependency $h \xrightarrow{l} d$ the label l represents the function binding the head h with the dependent d. Such a dependency is non-projective if at least one word located between the head and the dependent of the dependency does not depend on the head. Figure 1 presents an example of a non-projective dependency tree where the non-projective dependency connects a verb with a distant clitic.

The categorial dependency grammar of French [4], used to build the trees, has 116 different dependency labels. All dependencies with the same label describes specific information about the syntax of French. Most of the dependency labels can be gathered into larger syntactic groups describing more general information. For example, objects are separated into 7 dependency labels differentiating the grammatical cases (dative, accusative, etc.). Most of the dependency labels (89) are exclusively associated with projective dependencies. But some of them can be associated both with projective and non-projective dependencies. Among the

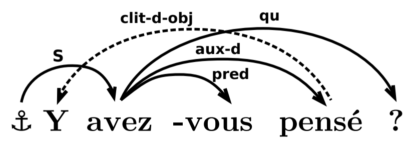

Fig. 1. Dependency tree for the sentence "Y avez-vous pensé ?" ("Did you think about it ?"). The clitic "y" ("it") depends on the verb "pense" ("think"). It refers to the indirect object of the verb (dative case).

23 dependency labels that can be combined with non-projective dependencies, the most frequent ones are clitics, negatives, objects, reflexives and copredicates. Four dependency labels are exclusively associated to non-projective dependencies, they are particular cases of aggregation, copula, comparison and negation.

The categorial dependency grammar of French, the non-projective dependency treebank and the parsing and treebank development environment that we use in this study is not yet publicly available, we have recovered them directly from the authors. In our work we use a treebank made up of sentences of various grammatical styles. A large part of these sentences (64%) were initialy used to develop the grammar of French. The whole corpus gathers several corpora composed of sentences from newspaper, 19^{th} and 20^{th} century literary works and plain language. We will call the joining of these treebanks the CDG Treebank. It is composed of 3 030 sentences (42691 words). Each sentence is paired with a dependency tree. The number of non-projective trees reaches 41.2% of the CDG Treebank. But, among all the dependencies, the non-projective ones represent 3.8% of all dependencies. The rate of non-projective dependencies varies from 1.3% to 4.9% according to the corpus[1].

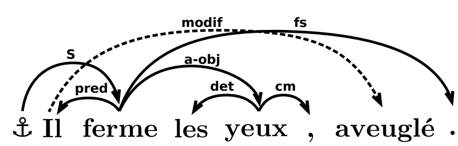

Fig. 2. Dependency tree for the sentence "Il ferme les yeux, aveuglé." ("He closes his eyes, blinded."). The modifier "aveuglé" ("blinded") depends on the subject "il" ("he").

---

[1] The sentences that were used to develop the grammar were choosed to cover all the syntactic phenomenon of French including the non-projective ones. Consequently, the rate of non-projective dependencies is more significant in the development corpus.

3.2 Annotation Process

The annotation process we propose includes four steps :

- sentence segmentation;
- automatic label prediction;
- dependency analysis;
- validation of the dependency tree.

The sentence segmentation is performed through a segmentation module which selects the longer lists of tokens recognizable by the lexical version of the CDGFr[2]. Here, we focus on the second step for which the role is to select the proper dependency label of each word[3]. Here, we call "the label of a word" the label of the dependency coming from the head to the word that is the dependent of the dependency. These labels correspond to the grammar's categories. For example, the labels of the words for the sentence used in Figure 2 are shown in Figure 3.

Il	**ferme**	**les**	**yeux**	**,**	**aveuglé**	**.**
pred	S	det	a-obj	cm	modif	fs

Fig. 3. The corresponding labels of the words for the sentence "Il ferme les yeux, aveuglé.".

Each word is also associated with a set of possible grammatical classes. The grammatical classes are extended tags (compared to classical part-of-speech tags) used by the CDGFr to categorize the words. The possible labels depend on the set of the possible classes of a word. Among these given possibilities, our goal is to select one or more labels for each word that are consistent with the categorial dependency grammar.The automatic procedure of this particular step is explained in detail in Section 4.

This pre-annotation step reduces the ambiguity of the next step, the grammar-based dependency analysis. Here, a CKY-based algorithm is applied to find all the possible dependency trees for the sentence. With the label pre-annotation, the analyser only considers the rules in adequacy with the selected categories (labels). This way, the number of generated dependency tree candidates greatly decreases. For example, the analysis of the sentence presented in Figure 2 generates 1518 (projective or non-projective) dependency trees without restrictions on labels. Selecting the proper labels reduces the number of possible dependency trees to 2.

Finally, the fourth step of the annotation process is validation. The task of the annotator involves annotating positively or negatively the dependencies of

[2] A recognizable list of tokens is a list constituting a lexical unit and not included in a black list which excludes some frequent errors of concatenation.

[3] We call words the possible combination of tokens that form a lexical unit. For instance, "Président Bill Clinton" has three tokens but corresponds to one word.

the resulting dependency tree and selecting the proper label and segmentation of the words for which a wrong label or segmentation was selected. Afterwards, a new analysis (iteration step) is performed taking account the annotations to approach the correct dependency tree (consistent with the grammar). This step can be performed as often as necessary and can include again the different steps of the pre-annotation process.

4 Sentence Pre-Annotation

Automatic label pre-annotation is the core of our annotation process and requires information about words and their grammatical context. Accordingly, we start by tagging the Part-Of-Speech tags.

4.1 POS-Tagging

The categorial dependency grammar makes use of 18 grammatical classes to categorize the words (e.g. noun, verb) and 10 for punctuation marks (e.g full stop, semicolon). The disjunction of some classes (e.g punctuation, particular verb types) is not necessary from a tagging point of view and can be ambiguous. Moreover, this tagset is not in adequacy with the tagset standardly used in French POS-taggers. Thus, in order to use a standard POS tagger and a tagset standardly used by the (French) community, we decided to convert our tagset into the TREEBANK+ tagset. This tagset consists in 28 tags extended from the classical tags used by the French Treebank [2], known to be efficient for parsing [16]. Furthermore, this tagset is used by MElt, a well studied French POS-tagger that achieves more than 97% accuracy on French [17].

Most of the grammatical classes correspond to TREEBANK+ tags, but some classes (e.g expletives, collocations, partitives) have no equivalent tags. These ones would make a direct conversion ambiguous. Therefore, we decided to conduct a mixed conversion. First, we tag automatically the whole corpus with the MElt tagger. Second, we correct the tags using basic rules for correction referring to the (non-ambiguous) original grammatical classes annotated in the CDG Treebank. The rate of correction on the tagset conversion reaches 6%. The most frequent errors are due to the ambiguity existing with adjectives acting as common nouns or past participle verbs acting as adjectives. Furthermore, some errors appear because of the differences between the sentences of the training corpus, a variant of the French Treebank [2], used by the MElt tagger and the sentences of the CDG Treebank [4]. The newly converted data are used in the label tagging experiments.

[4] One of the problems is that the training corpus contains very few imperative sentences and the CDG Treebank contains significantly more. Then, MElt is not able to find most of the imperative verbs. A lot of imperative verbs are tagged as indicative verbs. Others are tagged as nouns because this conjugated form are often located at the start of sentences with a first capital letter. and often tags the personal pronoun “tu” (“you”) as a verb because “tu” is also a conjugated form of the verb “taire” (“keep quiet”).

4.2 Label Pre-Annotation

Here, the goal is to find the labels but not the dependencies associated with the words. This automatic step should alleviate the work of the annotators. We need to use a rapid method to conduct the tagging. The parsing methods, trying to find both the label and the dependency, achieve equivalent scores (label accuracy) to these obtained by a method dedicated to tagging. However, we want to produce, for each word, a restricted list of the best labels with their probability scores. Therefore, among the probabilistic graphical models we choose the maximum entropy Markov model (MEMM) [18] to achieve this task because of its speed and the fact that the words are tagged independently[5].

To predict the labels, we try different combinations of features and test results. The features result in a combination of information from the lexical and grammatical context (a window size of 7 around the words and of 11 around the POS-tags). Then, we retrieve the 20 best labels for each word from the tagging. The list of labels is pruned from the labels which are not in the list of possible labels.

4.3 Label Sorting

Our model allows to keep control over the number of labels assigned to each word. In order to reduce the ambiguity, we want to eliminate the bad labels (i.e. the less probable ones) from the list of possible labels while preserving a high recall score. Each label (associated with a word) gets a probability score from the pre-annotation step. So, for a word, the idea is to eliminate the labels for which the probability score p_{max} is lower than $\alpha.p_{max}$ where p_{max} is the probability of the best label (the more probable one) and $\alpha \in [0, 1]$.

5 Experiments and Results

5.1 Experimental Settings

To evaluate the label pre-annotation process, we conduct a 10-fold cross-evaluation on the CDG treebank. Each experiment is performed on sentences POS-tagged with Melt.

To estimate the results, we calculate the precision of the label pre-annotation at rank one. It means, we calculate the percentage of words for which the first assigned label (i.e. the more probable) is the correct label. This precision corresponds to the label accuracy (LA) calculated on the output of a dependency parsing. Furthermore, we want to find a trade-off between increasing the recall on label accuracy and preserving a small number of labels per word. So, we evaluated the interest of the label sorting by varying the α parameter and connecting the recall with the number of labels assigned to each word.

[5] We use the software Wapiti [19] which is able to deal with a large tagset.

5.2 Results of the Label Pre-Annotation

Table 1 presents the results of the label pre-annotation. The scores do not reach the scores of projective dependency parsing of French that achieve more than 88% label accuracy. Actually, the scores are not comparable because of the constitution of the treebanks exploited in standard work. These commonly exploited French dependency treebanks, in addition to being projective, contain more sentences and use a smaller label set. Thus, to establish a baseline, we trained a transition-based parser, the MaltParser [6], on the sentences of the CDG Treebank. To exploit the potential of MaltParser we tested its available algorithms for non-projective dependency parsing. The best scores result from the use of the *covnonproj* algorithm and optimized features. Our label pre-annotation scores are slightly better than the label accuracy obtained from the data-driven dependency analysis.

Table 1. Evaluation of the label pre-annotation comparing our method with the performances of MaltParser.

	Label accuracy			**Sentence accuracy**		
	All	Proj.	Non-proj.	All	Proj.	Non-proj.
Our method	84.7	84.9	78.6	24.4	26.4	22.1
MaltParser	83.0	83.6	69.2	24.3	26.3	21.7

An interesting question is whether the non-projectivity affects the results. Table 1 also shows the accuracy of the label pre-annotation on the words for which a projective dependency ends on, in the original dependency tree, and for which a non-projective dependency ends on. The accuracy on words associated with a non-projective dependency achieves a lower score than for the words associated with a projective dependency. But, we note that our method achieves a better score than the MaltParser, on the words originally attached with a non-projective dependency. However, due to the small number of non-projective dependencies in the treebank (4%) the global score is weakly affected. The lower scores for words associated with non-projective dependencies can be explained by the fact that they are often attached by distant dependencies. This is not the case of current negation or clitization but some labels such as the aggregation or co-predication commonly attach distant words. Moreover, the context of distant dependents greatly differs from a sentence to another and thus cannot be learnt by the model.

Table 1 shows as well the accuracy on the sentences for both projective and non-projective ones. The non-projective dependencies do not represent a large part of the dependencies but are spread on many sentences (40%). So the effect of the non-projectivity on the accuracy on sentences is important.

A closer look shows that the best scores of accuracy among the different labels are achieved by the most frequent labels. They cover the most general syntactic function of French as subject, accusative object, determiners, modifiers, genitive prepositions. Likewise, the less frequent labels, describing very particular syntactic roles, are often subcategories of more general functions as the copulas, the auxiliaries, the object, etc. There are 34 labels appearing less than 20 times in the corpus which represents almost one third of the labels. These rare labels are not found at high ranks. This problem shows the importance of the label sorting. It allows to reach the second or more probable label for each word according to a given threshold. Figure 4 presents the results of the label sorting method as described in section 4.3. It highlights the progression of the recall according to the average number of more probable labels retrieved per word. For this experiment, the α parameter varies from 1 to 5.10^{-5}.

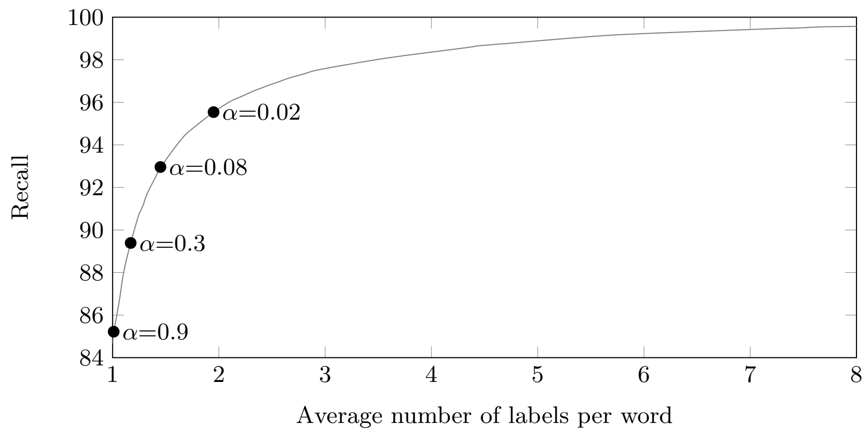

Fig. 4. Evaluation of the recall depending on the average number of labels found applying a sorting based on the probability scores.

5.3 Benefits of the Label Pre-Annotation on the Parsing Step

We evaluate the effect of the label pre-annotation on the parsing step of the annotation process (i.e. parsing with categorial dependency grammar). We present, in Table 2, the best parsing score we could obtain and the parsing time induced. The evaluation is performed on the CDG Treebank. The attachment scores are computed on the best dependency tree of each parses[6].

The first experiments are performed using the values of α indicated on Figure 4. For each experiment, the α parameter is fixed for the whole corpus. The last experiment is performed varying the α parameter according to the length of each parsed sentence. The longer the sentence, the higher (restrictive) α is.

[6] The best dependency tree is the tree having the most correct dependencies

The first threshold starts with $\alpha = 0.006$, allowing high recall scores for short sentences (< 10 words). Then, intermediate floors are defined until the last one ($\alpha = 0.9$) which preserves a small number of labels per word in order to parse the longest sentences (> 50 words) in reasonable time. Overall, defining progressive floors allows to find a trade-off between the parsing time and the attachment scores on the whole corpus. The goal is to conduct both a pre-annotation and a dependency parsing which are both accurate and not to long, in order to speed up the annotation process and alleviate the work of the annotator.

Table 2. Evaluation of the dependency parsing using the pre-annotation tool to assign one or more labels to each word. We present the best labelled attachment score (LAS) and the best unlabelled attachment score (UAS) that could be reached with this method.

Label sorting	Labels/words	Scores		Time (sec./sentence)
		LAS	UAS	
Fixed α	1.01	77.62	83.59	0.3
	1.17	81.10	86.47	0.8
	1.45	87.40	91.34	2.3
	1.95	91.94	94.62	7.2
Progressive α	2.04	90.16	92.89	3.0

We notice that the attachment scores increase slowly while the parsing time increases exponentially using a fixed α. The parsing time is decent for short sentences but explodes for long sentences when the number of pre-annotated labels per word is too large. The use of a progressive α is an interesting alternative which increases the attachment scores for short sentences (i.e. better chances to get the correct dependency tree increasing the label recall) and decreases the parsing time for the long ones (allowing to build at least one tree in a reasonable time).

6 Discussion about the Annotation Process

In order to estimate the impact of the pre-annotation step in the development of a dependency treebank we propose to annotate a small set of sentences from the different sub-corpora of the French treebank Sequoia [20]. We evaluate qualitatively the annotation process for two methods. The first one is the method using our automatic pre-annotation process, and the second one is the manual annotation process that uses word's label selection form. For a fair comparison of the methods, the annotation is performed on equivalent sentences (i.e. equivalent lengths).

The annotation of the sentences shows that our methodology is more suitable for the annotators. An advantage of the automatic pre-annotation is to skip the fastidious step of pre-selecting the labels. The annotators only have to validate the dependency trees. The benefit of the pre-annotation process is concrete on sentences of average and small length (<35) but minor on very long sentences. But overall, the average time saved with the first method is around half of the second.

Moreover, the assessment of the annotation highlights that some sentences of the Sequoia treebank are non-projective. The dependency annotation reveals the distant relations and the non-projective constructions that the constituent can not reveal. Around 28% of the annotated sentences have at least one non-projective dependency.

7 Conclusion and Future work

We show that the scores of a label tagging method using a maximum entropy Markov model are equivalent to the label accuracy scores obtained with a standard data-driven dependency parser. These scores do not reach the scores reported in works on projective dependency parsing because finding the non-projective dependencies is a difficult task. However, the method reaches interesting recall scores which allow to retrieve the right labels while keeping control over the ambiguity reduction. Consequently, this automatic pre-annotation tool included in the whole annotation process relieves the work of the annotators. Part of the time is saved and the annotation process is more accessible. Avoiding the pre-annotation step is greatly appreciated even if the validation step requires some corrections.

Moreover, the evaluation of dependency parsing using the pre-annotation tool shows that we could obtain good scores on non-projective dependency parsing. We plan to improve the sorting of the dependency trees in order to propose a complete parser which is able to deal with non-projective constructions and reach appropriate scores.

References

1. Kübler, S., McDonald, R., Nivre, J.: Dependency Parsing. Synthesis Lectures on Human Language Technologies **1**(1) (2009) 1–127
2. Abeillé, A., Clément, L., Kinyon, A.: Building a Treebank for French. In: Proceedings of the Language Resources and Evaluation Conference. LREC 2000, Athens, Greece (May 2000)
3. Candito, M., Crabbé, B., Denis, P.: Statistical French Dependency Parsing: Treebank Conversion and First Results. In: Proceedings of the Language Resources and Evaluation Conference. LREC 2010, Valletta, Malta (May 2010)
4. Dikovsky, A.: Categorial Dependency Grammars: from Theory to Large Scale Grammars. In: Proceedings of the International Conference on Dependency Linguistics. DEPLING 2011 (September 2011)
5. Alfared, R., Béchet, D., Dikovsky, A.: CDG Lab: a Toolbox for Dependency Grammars and Dependency Treebanks Development. In: Proceedings of the International Conference on Dependency Linguistics. DEPLING 2011, Barcelona, Spain (September 2011) 272–281

6. Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kbler, S., Marinov, S., Marsi, E.: MaltParser: A Language-Independent System for Data-Driven Dependency Parsing. Natural Language Engineering **13** (6 2007) 95–135
7. Buchholz, S., Marsi, E.: CoNLL-X Shared Task on Multilingual Dependency Parsing. In: Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X), New York City, Association for Computational Linguistics (June 2006) 149–164
8. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a Large Annotated Corpus of English: the Penn Treebank. Comput. Linguist. **19**(2) (June 1993) 313–330
9. Hajič, J., Hajičová, E., Pajas, P., Panevová, J., Sgall, P., Hladká, B.V.: Prague Dependency Treebank 1.0 (Final Production Label) (2001)
10. Green, N., Larasati, S.D., Žabokrtský, Z.: Indonesian Dependency Treebank: Annotation and Parsing. In: Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation. PACLIC 2012, Bali, Indonesia (November 2012) 137–145
11. McGillivray, B., Passarotti, M., Ruffolo, P.: The Index Thomisticus Treebank Project: Annotation, Parsing and Valency Lexicon. TAL **50**(2) (2009) 103–127
12. Atalay, N.B., Oflazer, K., Say, B., Inst, I.: The Annotation Process in the Turkish Treebank. In: Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora. LINC-03, Budapest, Hungary (April 2003)
13. Cerisara, C., Gardent, C., Anderson, C.: Building and Exploiting a Dependency Treebank for French Radio Broadcast. In: Proceedings of the ninth international workshop on Treebanks and Linguistic Theories. TLT9, Tartu, Estonia (November 2010)
14. Alfared, R., Béchet, D.: On the Adequacy of Three POS Taggers and a Dependency Parser. In: Proceedings of the 13th international conference on Computational Linguistics and Intelligent Text Processing - Volume Part I. CICLing'12, New Delhi, India (March 2012) 104–116
15. Bangalore, S., Joshi, A., eds.: Complexity of Lexical Descriptions and its Relevance to Natural Language Processing: A Supertagging Approach. MIT Press (2010)
16. Crabbé, B., Candito, M.: Expériences d'analyse syntaxique statistique du français. In: Proceedings of the Joint Conference JEP-TALN-RECITAL 2008, Avignon, France (2008)
17. Denis, P., Sagot, B.: Coupling an Annotated Corpus and a Morphosyntactic Lexicon for State-of-the-Art POS Tagging with Less Human Effort. In: Proceedings of the Pacific Asia Conference on Language, Information and Computation. PACLIC 2009, Hong Kong, China (2009)
18. Ratnaparkhi, A.: A Maximum Entropy Model for Part-Of-Speech Tagging. In: Proceedings of the 1st Conference on Empirical Methods in Natural Language Processing. EMNLP 1996, Pennsylvania, USA (May 1996)
19. Lavergne, T., Cappé, O., Yvon, F.: Practical Very Large Scale CRFs. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. ACL 2010 (July 2010) 504–513
20. Candito, M., Seddah, D.: Le corpus Sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical (The Sequoia Corpus : Syntactic Annotation and Use for a Parser Lexical Domain Adaptation Method) [in French]. In: Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 2: TALN, Grenoble, France (June 2012) 321–334