Validation Issues Induced by an Automatic Pre-Annotation Mechanism in the Building of Non-projective Dependency Treebanks

Ophélie Lacroix, Denis Béchet

LINA - University of Nantes
2 Rue de la Houssinière, BP 92208, 44322 NANTES CEDEX 3, FRANCE
ophelie.lacroix@univ-nantes.fr, denis.bechet@univ-nantes.fr

Abstract

In order to build large dependency treebanks using the CDG Lab, a grammar-based dependency treebank development tool, an annotator usually has to fill a selection form before parsing. This step is usually necessary because, otherwise, the search space is too big for long sentences and the parser fails to produce at least one solution. With the information given by the annotator on the selection form the parser can produce one or several dependency structures and the annotator can proceed by adding positive or negative annotations on dependencies and launching iteratively the parser until the right dependency structure has been found. However, the selection form is sometimes difficult and long to fill because the annotator must have an idea of the result before parsing. The CDG Lab proposes to replace this form by an automatic pre-annotation mechanism. However, this model introduces some issues during the annotation phase that do not exist when the annotator uses a selection form. The article presents those issues and proposes some modifications of the CDG Lab in order to use effectively the automatic pre-annotation mechanism.

Keywords: Non-projective Dependency Treebank, Treebank Development Tool, Pre-annotation

1. Introduction

The most known available dependency treebanks for French are converted from constituency treebanks (Abeillé et al., 2000; Candito and Seddah, 2012). The dependency trees generated from the conversion are projective and they are not able to represent some non-projective relations due to discontinuous grammatical structures of French.

The CDG Lab (Alfared et al., 2011; Béchet et al., 2014), an integrated environment for the development of dependency grammars and treebanks, proposes to create dependency treebanks containing both projective and non-projective dependency trees. Even if the main purpose of the CDG Lab is the parallel construction of a Categorial Dependency Grammar (Dekhtyar and Dikovsky, 2008) and a treebank in accordance with the grammar, the environment can be used to define a new treebank compatible to a given grammar.

For the construction of a dependency tree when the grammar is fixed, the environment includes a parser whose architecture is shown in Figure 1. An input form is filled with the sentence that must be added to the treebank. Then either the parser is launched directly (autonomous analysis) or a selection form is first displayed (analysis by head dependency selection). This second method is usually necessary with long sentences and a large scale grammar because without a selection, the search space is too big and the parser fails to propose at least one solution.

For a sentence, the selection consists in choosing the proper lexical units, their grammatical classes and head dependency labels through the selection form. Figure 2 shows an example for "*Pierre boit de la bière belge*" (Peter drinks some Belgium beer) where the possible lexical units are associated to a list of grammatical classes and a list of dependency labels (or dependency groups). The dependency label of a lexical unit is the label of the dependency that ends in the unit.

The selection defined by the annotator reduces the ambiguity of the grammar-based dependency parsing. However, selection forms are long and difficult to fill. Annotators usually prefer to work directly on dependency trees. In order to improve the annotators comfort and their productivity, (Béchet et al., 2014) proposed to modify the autonomous analysis by the introduction of an oracle that performs an automatic selection of the lexical units and their head dependency label. Thus, the automation of the pre-annotation process allow to move quickly to the validation step.

Even if the pre-annotation mechanism is relatively precise, some errors with long sentences indicates that the principle of analysis by approximation in Figure 1 sometimes fails to produce the right dependency tree. The paper shows examples of such failures and presents a modified parser tool that enables every annotator to bypass those problems and find the proper dependency tree during a modified "analysis by approximation".

2. Related Work

The most known and used treebanks for French (Abeillé et al., 2000; Candito and Seddah, 2012) were built manually or semi-automatically and are constituency treebanks. The dependency version of these treebanks were converted from the constituency ones. Consequently, the dependency structures are all projectives and their constructions do not lead to the same issues because the sentences were previously well segmented and POS tagged.

Overall, for other languages, the methodology of construction often includes the use of a parser. For example, to build the dependency treebank for Russian (Boguslavsky et al., 2002), the Turkish treebank (Atalay et al., 2003), the Prague Dependency Treebank (Hajič et al., 2000), the annotators previously used a data-driven parser (e.g MaltParser (Nivre et al., 2007)). Otherwise, the building of treebank

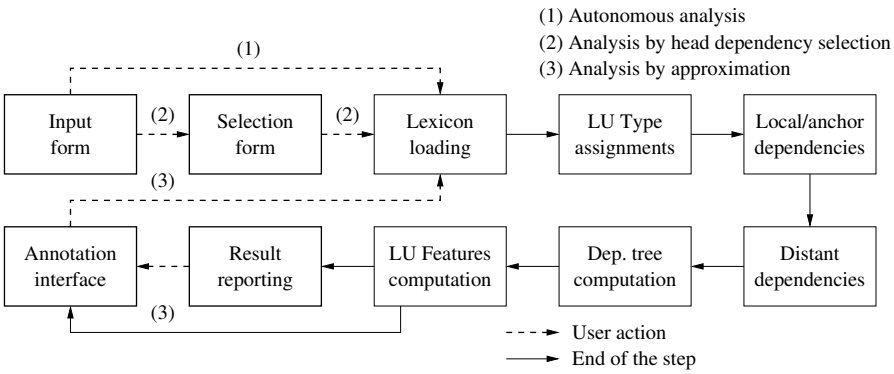

Figure 1: Architecture of the CDG Lab Parser.

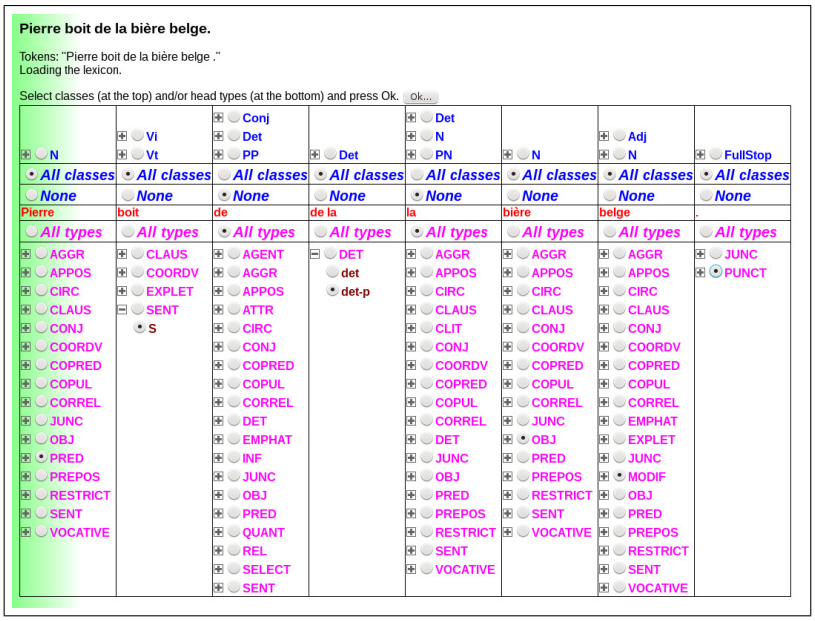

Figure 2: Selection Form (screenshot).

can be carry out with the help of a grammar-based parser (associated with statistical processes or not) as the LFG (lexical functional grammar) for TIGER treebank (Brants et al., 2002) or the DEPBANK (King et al., 2003), or the constraint grammar (Karlsson et al., 1995) for the Arboretum treebank (Bick, 2003).

In our case, we aim at building a treebank for which the dependency trees are consistent with a grammar. Thus, we use a grammar-based parser to provide the dependency structures. The inclusion of a data-driven pre-annotation in the process is a trade-off between the speed of the annotation process and the consistency with the grammar.

3. Pre-Annotation

3.1. The Manual Pre-Annotation Process

The manual pre-annotation process consists in one large selection form as the one presented in Figure 2. An annotator selects some grammatical classes and head dependency labels. These selections induce the selection of lexical units. The classes and the head dependency labels can be selected among restricted lists. The list of possible grammatical classes is given by a lexicon included in the grammar. For instance, the grammar for French (Dikovsky, 2011) called here *CDGFr* that we use for our experiments is based on information of Lefff (Sagot, 2010).

3.2. The Automatic Pre-Annotation Process

The automatic pre-annotation process consists in a segmentation into lexical units and a prediction of their POS-tags and dependency labels. It is the preliminary step in our work but does not constitute the main topic of this paper. Thus, for this part, the experiments were performed on a French dependency treebank which includes both projective and non-projective dependency structures. This treebank, called in this paper as the CDG treebank, consists of 3030 sentences. The results are presented in this section to appraise the efficiency of this step while the validation issues are discussed in the next section.

The segmentation module used a very simple algorithm that selects the longest list of tokens recognized by the parser of the CDG Lab using the selected grammar. This simple version has a relatively high level of errors (2.85%) when

tested on the CDG treebank. In fact, a lot of them come from a very small number of words and a black list of lexical units has been added to the first algorithm. For example some errors come from the case of partitive articles. In French, the segment "*de la*" is always considered as the feminine equivalent of "*du*" (contraction of "*de*" and "*le*") when it is not often the case. The black list (13 cases at all) does not correct all the segmentation problems but allows to achieve a decent score of 1.62% error.

The POS tagging and the dependency label tagging are data-driven processes. The training data, from the CDG treebank, were annotated with a set of 28 grammatical classes which is not standard but consistent with the CDGFr. Our experiments on POS tagging use MElt (Denis and Sagot, 2009), a POS-tagger for French. The tagger achieves high accuracy on French and the tagset is known to be efficient for parsing (Crabbé and Candito, 2008). Thus, the tagset of the CDG treebank were converted into the tagset of Melt.

The evaluation on POS-tagging was performed through a 10-fold cross-evaluation and the results are presented in Table 1. The precision on tokens is lower than the baseline of French POS tagging. The main wrong assignations are due to conversion errors when there is no equivalence between the grammatical classes and the POS tags. Moreover, the tagsets were initially established from different linguistic theories. It results a lack of precision during the conversion. Also, some errors result from the differences between the source used for the training corpus (a variant of the French treebank (Abeillé et al., 2000)) and the test corpus (the CDG treebank). For example, the training corpus were built from newspapers which do not include imperative sentences when our corpus contains significantly more.

	Precision	**Std. dev.**
Tokens	92.54	0.26
Sentences	40.17	2.27
Unknown words	88.98	0.15

Table 1: *Evaluation of the POS-tagging: precision and standard deviation.*

The label tagging serves only to find the head dependency labels and not the dependencies associated with the lexical units. A maximum entropy Markov model (MEMM) (Ratnaparkhi, 1996) was chosen for this step. The method uses information about lexical units and their POS-tags to predict the head dependency labels. In addition, the predicted labels are sorted and the better label that is also proposed by the parser (in the list of possible labels) is selected. The results of the label tagging are presented in Table 2.

	Precision	**Ratio**
Lexical units	83.55	
Projective	83.88	96%
Non-projective	76.33	4%
Sentences	21.23	
Projective	22.67	59%
Non-projective	19.53	41%

Table 2: *Evaluation of the label tagging. We differentiate the lexical units on which a projective or a non-projective dependency ends in the original dependency tree. We also differentiate the projective and non-projective sentences. A sentence is considered as non-projective when at least one non-projective dependency appears in the original dependency tree.*

4. Issues with Parsing by Approximation

Pre-annotation replaces the selection performed by an annotator through the selection form. Even if the step is precise, it cannot be perfect and some errors are introduced that have sometimes wrong consequences during the phase on which the annotator tries to find the right dependency tree among the solutions proposed by the parser. If all dependencies are correct in the dependency tree proposed by the parser, the annotator puts the dependency tree in the treebank and continues with the next sentence. Otherwise, with the CDG Lab, the annotator annotates positively or negatively each dependency that appears on the dependency tree proposed by the parser. Then he/she launches the parser again. The parser proposes a better solution: A solution that has less negatively annotated dependencies and more positively annotated dependencies. This mechanism is called "Parsing by Approximation" on Figure 1.

4.1. No Dependency Tree

With an automatic pre-annotation phase, the selection of lexical units and head dependency labels can lead to a search space that has no compatible solution to the grammar. In this case, the parser gives no dependency tree and the annotator cannot continue: The search space is too small.

This problem has been partially resolved in the new version of the CDG Lab. The parser can produce partial dependency trees when a complete solution is not possible. This structure is a list of dependency trees corresponding to the parsing of sub-parts of the initial sentence. This first solution is presented to the annotator which can proceed to parsing by approximation. Figure 4 shows an example of a partial solution which contains 3 dependency trees.

4.2. Wrong Lexical Units

With an automatic pre-annotation phase, the annotator cannot choose the right lexical units and, for each of them, the right head dependency label. As a consequence, the lexical unit segmentation can be wrong. For instance, in Figure 2, the pre-annotation mechanism could have chosen the lexical units "*de*" (preposition) and "la" (determiner) rather than "*de la*" (partitive determiner). As a consequence, the partitive determiner is disable during parsing and will not be used in the solutions proposed by the parser. Figure 4 shows a partial solution given by the parser that is a consequence of a wrong segmentation of "*de la*".

For this problem, a new annotation mechanism has been

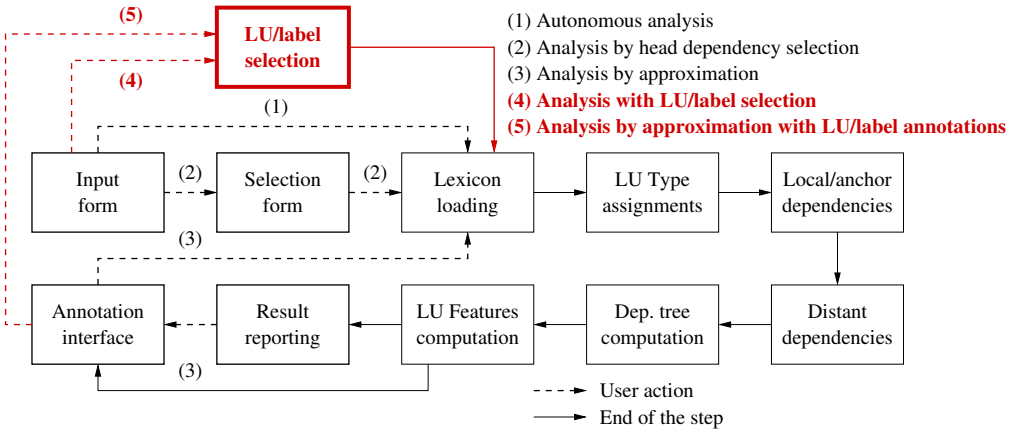

Figure 3: New architecture of the CDG Lab Parser.

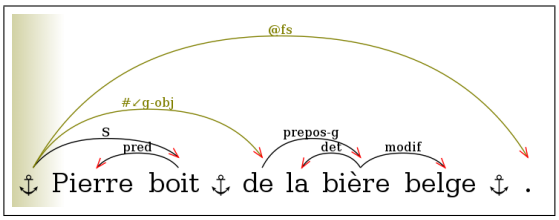

Figure 4: Partial solution (screenshot).

introduced. The annotator can select the right lexical unit corresponding to a word using a contextual menu. For instance, with the example of Figure 4, "*de*" is associated to "*de*" and "*de la*". "*la*" is associated to "*de la*" and "*la*". The annotator can select "*de la*" and launches the parser. The new solution proposed by the parser has now a correct segmentation. Figure 5 shows the solution proposed after the annotation of the correct dependencies of Figure 4 and the selection of "*de la*" in place of "*de*" and "*la*".

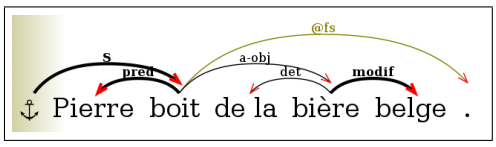

Figure 5: Next approximation (screenshot).

4.3. Wrong Head Dependency Labels

The pre-annotation tool selects a dependency label for each lexical unit. The parser only shows solutions where the lexical unit is the end (the argument) of a dependency whose label is the one that is selected for this unit.

The label may be wrong. In this case, the annotator can only annotate negatively the dependency that ends in the lexical unit. The parser only knows that the negatively annotated dependencies are not correct. It cannot infer that all the dependencies with that label are not correct. To solve this problem, the annotator must be able to indicate that a label chosen for a unit during pre-annotation is not correct. Our solution to this problem consists in providing, in the contextual menu associated to a lexical unit, the list of labels computed during pre-annotation. The parser then can produce solutions where the lexical unit has the right head dependency label.

5. Discussion and Conclusion

The CDG Lab's analysis by head dependency selection used for the creation of dependency treebank depends on a selection form before parsing that is long and difficult to fill by annotators.

An efficient automatic pre-annotation mechanism presented in (Béchet et al., 2014) has been added to the environment. Our first experiments to create new corpora based on the sentences of the Sequoia treebank (Candito and Seddah, 2012) shows that the annotators save approximatively 25% of time using automatic pre-annotation rather than using the manual selection form. However, the experiment also shows that the annotators have used special non natural strategies during the parsing by approximation. Because pre-annotation introduces errors, the annotators annotate temporarily some false dependencies as correct in order to work in other part of the sentence with a not too big search space. This strange strategy is a curious response of the annotators to the problem we have presented in the previous section. The solution that we propose here implements the principle of automatic pre-annotation as a replacement of the selection form but it is much more natural to annotators that the mechanism proposed in (Béchet et al., 2014).

The modifications that we have introduced are now included in the new main version of the CDG Lab. The new architecture of the CDG Lab Parser is shown in Figure 3. A new module has been added that selects lexical units (LU) and their head dependency labels (label) using the automatic pre-annotation process. This module can be called from the input form when a new sentence is parsed (analysis by LU/label selection) or from the annotation interface when a previous annotated analysis exists (analysis by approximation with LU/label annotations). In this case, the new module selects lexical units and head dependency labels with regard on the annotations entered by the user with the annotation interface (i.e. the right lexical unit and its head dependency label). Some modules have been

modified. In particular, the dependency trees of the annotation interface can show the list of lexical units that begin at the same position of the current one. The interface can also show the list of possible head dependency labels of each lexical unit. The user can select an alternative lexical unit (LU annotation) and one of the head dependency label (label annotation). The module that computes dependency trees has been changed in order to compute a forest of dependency trees when no complete dependency tree exists.

6. Acknowledgement

We want to thank Danièle Beauquier and Alexandre Dikovsky for giving us the CDG of French (Dikovsky, 2011) (called here CDGFr) and the corpora they have developed. Without them, it would be impossible to test the CDG Lab and propose these enhancements for the creation of new corpora.

7. References

Anne Abeillé, Lionel Clément, and Alexandra Kinyon. 2000. Building a treebank for French. In *Proceedings of the Language Resources and Evaluation Conference*, LREC 2000, Athens, Greece, May.

Ramadan Alfared, Denis Béchet, and Alexander Dikovsky. 2011. CDG Lab: a toolbox for dependency grammars and dependency treebanks development. In *Proceedings of the International Conference on Dependency Linguistics*, DEPLING 2011, pages 272–281, Barcelona, Spain, September.

Nart B. Atalay, Kemal Oflazer, Bilge Say, and Informatics Inst. 2003. The annotation process in the Turkish treebank. In *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora*, LINC-03, Budapest, Hungary, April.

Denis Béchet, Alexander Dikovsky, and Ophélie Lacroix. 2014. "CDG Lab": an integrated environment for categorial dependency grammar and dependency treebank development. In Kim Gerdes, Eva Hajičová, and Leo Wanner, editors, *Computational Dependency Theory*, volume 258 of *Frontiers in Artificial Intelligence and Applications*, pages 153–169. IOS Press.

Eckhard Bick. 2003. Arboretum, a hybrid treebank for Danish. In *Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories*.

Igor Boguslavsky, Ivan Chardin, Svetlana Grigorieva, Nikolai Grigoriev, Leonid L. Iomdin, Leonid Kreidlin, and Nadezhda Frid. 2002. Development of a dependency treebank for Russian and its possible applications in NLP. In *In Proceedings of the 3rd International Conference on Language Resources and Evaluation, Las Palmas, Gran Canaria*.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank.

Marie Candito and Djamé Seddah. 2012. 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*, pages 321–334, Grenoble, France, June.

Benoît Crabbé and Marie Candito. 2008. Expériences d'analyse syntaxique statistique du français (statistical parsing of French) [in French]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2008*, Avignon, France.

Michael Dekhtyar and Alexander Dikovsky. 2008. Generalized categorial dependency grammars. In *Trakhtenbrot/Festschrift*, LNCS 4800, pages 230–255. Springer.

Pascal Denis and Benoît Sagot. 2009. 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.

Alexander Dikovsky. 2011. Categorial dependency grammars: from theory to large scale grammars. In *Proceedings of the International Conference on Dependency Linguistics*, DEPLING 2011, September.

Jan Hajič, Alena Böhmová, Eva Hajičová, and Barbora Vidová-Hladká. 2000. The Prague Dependency Treebank: A Three-Level Annotation Scenario. In A. Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, pages 103–127. Amsterdam:Kluwer.

Fred Karlsson, Atro Voutilainen, Juha Heikkila, and Atro Anttila. 1995. *Constraint Grammar, A Language-independent System for Parsing Unrestricted Text*. Mouton de Gruyter.

Tracy Holloway King, Richard Crouch, Stefan Riezler, Mary Dalrymple, and Ronald M. Kaplan. 2003. The PARC 700 dependency bank. In *In Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC-03)*, pages 1–8.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Glsen Eryigit, Sandra Kbler, Svetoslav Marinov, and Erwin Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13:95–135, 6.

Adwait Ratnaparkhi. 1996. 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.

Benoît Sagot. 2010. The Lefff, a freely available and large-coverage morphological and syntactic lexicon for French. In *Proceedings of the Language Resources and Evaluation Conference*, LREC 2010.