Complex Function Words in French: Representation and Processing

Alexis Nasr and **José Deulofeu** and **André Valli** and **Carlos Ramisch**
Aix Marseille Université, CNRS, LIF UMR 7279
13288, Marseille, France
FirstName.LastName@lif.univ-mrs.fr

1 ADV+*que* Conjunctions

Function words are lexical units with generally little semantic weight that often play a role of "grammatical" elements in a sentence, introducing or modifying content words. These include prepositions (*with*), determiners (*some*), pronouns (*she*) and conjunctions (*furthermore*). *Complex function words* are function words made up of several tokens, like complex prepositions (*in front of*), determiners (*a lot of*) and conjunctions (*as long as*).

This abstract discusses the representation and detection of ADV+*que* constructions, a type of complex conjunction in French. These constructions are formed by adverbs like *bien* (*well*) or *ainsi* (*likewise*) followed by subordinative conjunction *que* (*which*).

Due to their structure, ADV+que constructions are generally ambiguous such as in the following sentences:

1. *Je mange* **bien que** *je n'aie pas faim*
   I eat **although** *I am not hungry*

2. *Je pense* **bien que** *je n'ai pas faim*
   I **really think** *that I am not hungry*

In sentence 1, the sequence *bien que* forms a complex conjunction whereas in sentence 2, the adverb *bien* modifies the verb *pense* and the conjunction *que* introduces the sentential complement *je n'ai pas faim*.

We consider a standard NLP analysis pipeline made of three modules: tokenizer, part-of-speech tagger and syntactic parser. The question that occurs is: which module is responsible for detecting the occurrence of complex conjunctions? It is often assumed that the tokenizer must perform this task and represents as a single lexical unit the ADV+*que* conjunction.

This situation is not satisfactory since such ambiguous constructions can only be detected based on syntactic clues: the fact that the verb of the principal clause accepts a *que* complement and the mood of the verb of the subordinate clause. Tokenizers are usually simple software which use regular expressions and lists of tokens to split the input. They do not usually have access to syntactic or morphological information.

Parsers are therefore in a better position to detect such construction. But parsers usually do not associate a syntactic structure to such structures and consider them as a single lexical unit, using a *words-with-spaces* approach.

In order to solve the problem, tokenizers can produce several tokenizations of the sentence. In our case, we will have a tokenization in which the complex conjunction has been recognized as a single token and another one in which the adverb and the subordinative conjunction are represented as independent tokens. The two tokenizations can be represented as two separate sequences of tokens or as a directed acyclic graph, in which common parts of the sequences have been represented only once, such as in (Nasr et al., 2011). All tokenizations are sent to the tagger and then to the parser.

Although this approach will produce different syntactic structures, the parser will have a hard time for selecting the most likely one. Indeed, different paths in the tokenization graph will be made of a different number of tokens and the scores of the parses produced for the different tokenizations, on the basis of which the choice for the most likely structure is done, will be difficult to compare.

2 The MORPH dependency

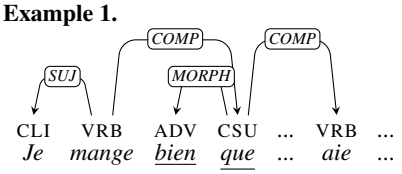
In order to solve this problem we propose *not* to group sequences of tokens that can form complex conjunctions at tokenization time. More precisely, we only concatenate as a single token multiword elements containing lexical elements that never occur in isolation. For example, the word *parce* has no sense in itself, and only occurs in *parce que* (*because*). Therefore, it makes sense joining

these units at tokenisation time and treating them as single token in succeeding steps. However, the word *lors* only occurs in *dès lors* (*since then*) and *lors de* (*at the time of*), but can be ambiguous if it occurs in *dès lors de*. We solve this by always concatenating the leftmost words.

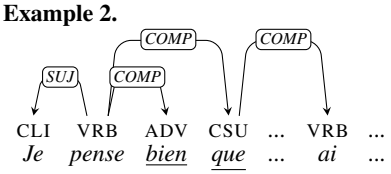
To enable the parser to associate a syntactic structure to complex conjunctions, we introduce a new relation type that we call MORPH. It is not a standard dependency, but a reminiscent of the morphological dependencies of Mel'čuk (1988). It is similar to the *dep_cpd* label proposed by Candito and Constant (2014), except that we focus on a specific MWE type.

The syntactic structure of the two sentences introduced in section 1 are represented below.

Example 1.



Example 2.



In sentence 1, the complex conjunction *bien que* is represented by the presence of the MORPH dependency, whereas, in sentence 2, the adverb *bien* modifies the verb *pense*.

From an NLP perspective, the two readings of *bien que* are treated the same way by the tokenizer and the tagger. It is only at parsing time that the presence of the complex conjunction is predicted.

3 Experiments

In order to test our idea, we have extracted from the French web as a corpus (Baroni et al., 2009) the eight most frequent ADV+*que* constructions. 100 sentences for each of them have been randomly selected and manually labelled. The label indicates for each sentence if the ADV+*que* construction is a complex conjunction or not. The results are represented in table 1. The second and third columns indicate respectively the number occurrences for which the ADV+*que* construction forms a complex conjunction or not.

Table 1 shows that, contrary to our initial intuition, ADV+*que* constructions only show a slight tendency to form complex conjunctions. As can be seen in the table, different ADV+*que* constructions exhibit different behaviours. Some of them, such as *depuis que*, form almost always a complex conjunction while *tant que* or *encore que* form a complex conjunction in only 20% of the cases.

ADV+que	comp. conj.	other
ainsi que	0.76	0.24
alors que	0.88	0.12
autant que	0.86	0.14
bien que	0.37	0.63
encore que	0.21	0.79
maintenant que	0.57	0.43
tant que	0.20	0.80
total	0.56	0.44

Table 1: The 7 most frequent ADV+que structures and their ambiguity

In order to train a parser to predict the MORPH dependency, we have modified the French Treebank (Abeillé et al., 2003) annotation scheme in such a way that the ADV+*que* construction that appear as complex conjunctions, which are represented in the French Treebank as single tokens, are represented as two tokens linked with the MORPH dependency.

We used a second-order graph-based parser (Kübler et al., 2009). It has been evaluated on the manually annotated sentences extracted from the French web as a corpus. Our evaluation concentrates on the prediction of the MORPH dependency. Recall, precision and f-measure for the eight ADV+*que* constructions are shown in table 2. These first results show that the precision is quite good, comparable to the average precision of standard syntactic dependencies. But recall is quite low.

ADV+que	recall	prec.	f-meas.
ainsi que	0.96	0.95	0.95
alors que	0.93	0.94	0.93
autant que	0.54	0.92	0.68
bien que	0.84	0.86	0.85
encore que	0.80	0.84	0.82
tant que	0.75	1.00	0.86
maintenant que	0.75	0.89	0.81
total	0.80	0.91	0.84

Table 2: Precision, recall and f-measure of the prediction of the MORPH dependency

References

A. Abeillé, L. Clément, and F. Toussenel. 2003. Building a treebank for french. In Anne Abeillé, editor, *Treebanks*. Kluwer.

M. Baroni, S. Bernardini, A. Ferraresi, and E. Zanchetta. 2009. The wacky wide web: A collection of very large linguistically processed web-crawled corpora. volume 43, pages 209–226.

Marie Candito and Matthieu Constant. 2014. Strategies for contiguous multiword expression analysis and dependency parsing. In *Proc. of the 52nd ACL (Volume 1: Long Papers)*, pages 743–753, Baltimore, MD, USA, Jun. ACL.

S. Kübler, R. McDonald, and J. Nivre. 2009. Dependency parsing. *Synthesis Lectures on Human Language Technologies*, 1(1):1–127.

Igor A. Mel'čuk. 1988. *Dependency Syntax: Theory and Practice*. State University of New York Press, New York.

Alexis Nasr, Frederic Bechet, Jean-Francois Rey, Benoit Favre, and Joseph Le Roux. 2011. MACAON an NLP tool suite for processing word lattices. In *Proc. of the ACL 2011 System Demonstrations*, pages 86–91, Portland, OR, USA, Jun. ACL.