A Case Study in Implementing Dependency-Based Grammars

Marie BOURDON, Lyne DA SYLVA, Michel GAGNON,
Alma KHARRAT, Sonja KNOLL, Anna MACLACHLAN
Les Logiciels Machina Sapiens inc.
3535, chemin de la Reine-Marie, bureau 420,
Montréal, QC,
Canada
mbourdon,ldasylva,mgagnon,akharrat,sknoll,amaclachlan@machinasapiens.com
http://www.machinasapiens.com

Abstract

In creating an English grammar checking software product, we implemented a large-coverage grammar based on the dependency grammar formalism. This implementation required some adaptation of current linguistic description to prevent serious overgeneration of parse trees. Here, we present one particular example, that of preposition stranding and dangling prepositions, where implementing an alternative to existing linguistic analyses is warranted to limit such overgeneration.

Introduction

Implementing a linguistic theory such as dependency grammar leads to many types of problems (see the discussion in Fuchs *et al*, 1993, p.121ff, among others). We will focus on a problem typical of large-scale NLP implementations: some theoretical descriptions entail unforeseen computational costs.

The linguistic phenomenon chosen to illustrate this problem is the case of so-called stranded and dangling prepositions in English. We will show how our initial description, akin to those presented in the dependency grammar literature, led to inefficiency in the parser. By modifying the grammatical analysis in some cases, rather than the parser itself, an overall improvement was achieved.

This problem raises the issue of the difficulties inherent in the large-scale implementation of a theoretical grammar which has been designed to describe linguistic phenomena and not as the basis of a parser.

1 An implementation of a broad-coverage dependency-based grammar

The grammar constitutes the backbone of our grammar checker software for different languages, including French, Spanish, English and Portuguese. Our checkers belong to the third generation of such products, which perform a complete and detailed grammatical analysis of sentences.

A commercially viable grammar checker must catch all the errors in a text and only those errors. Crucially, it must do so in a relatively short time on moderately powerful machines. Performing an accurate linguistic analysis of texts requires time and appropriate strategies. Some developers avoid the problems associated with performing a complete grammatical analysis by using local (or semi-local) methods of processing instead. It seems obvious, however, that the more linguistic knowledge a checker has, the better its chances of identifying errors.

Our grammar checker, which performs a complete linguistic analysis of all sentences, is based on a dependency grammar. This type of grammar was originally perceived as being intuitively more efficient in computational terms, allowing simple descriptions that can be parsed in an incremental manner. It has indeed proved to be efficient in our implementation.

Some of that efficiency is due to the initial constraints placed on the system, including the following among others. First, every word in the input sentence corresponds to a node in the structure built (with minor exceptions). Second, only adjacent subtrees may be combined. Third, each node may have at most one father. These restrictions also limit the types of linguistic analyses we can implement, as we will illustrate.

1.1 The grammar checker software

The coverage and accuracy of the linguistic descriptions on which our grammar checker is based determine its strength. Grammatical structures that are difficult to describe and explain are not necessarily considered by the layman as being particularly problematic (take for example coordination). Moreover, a commercial product cannot survive if it fails to treat some cases that are obvious to a user. For example, punctuation falls outside the description provided by most syntactic theories but it is pervasive in writtent texts and must be handled and perhaps corrected.

Our grammar checker is aimed at the general public and is designed to analyze written texts from a range of domains and in a range of styles. It therefore requires a grammar with a very broad coverage as well as a very extensive lexicon.

It is implemented in C++. The English lexicon consists of 65,000 English root words. Syntactic structures handled by the parser include the core of grammar (noun groups, verb groups, prepositional groups, etc.) as well as: declaratives, interrogatives, relatives, exclamatives, imperatives, comparatives, superlatives, most coordinate structures, many elliptical structures, punctuation, constructions belonging to the grammar of correspondence, such as addresses, and some types of idiomatic expressions.

The grammar checker proceeds in the following way. It starts by performing a lexical analysis. Some phonetic approximation rules may be used to deal with unrecognized words or to resolve incomplete parses. The syntactic component uses a set of dependency rules, which involve some simplification of the structures postulated within the literature on dependency grammar. Once an analysis is generated, grammatical corrections are performed and the result is displayed to the user.

Theoretical approaches modelling how humans parse language often start or finish with a semantic representation. Our parser, however, deals only with surface structure. There is no semantic component to our product *per se*, but a small number of semantic features are used. Given the commercial success of our grammar checker, it can be considered a successful implementation.

1.2 A central problem

One of the key problems in implementing an NLP system is dealing with combinatorial explosion: in attempting to produce the analysis for a sentence given a potentially very large set of rules, some strategies must be used to reduce the search space. Otherwise the time necessary to complete the computation may be too long.

We will not exhaust the types of difficulties that were encountered and solved, but will focus primarily on one problem stemming from a linguistic analysis which entailed the creation of a large search space: stranded and dangling prepositions.

2 A problematic phenomenon: stranded and dangling prepositions

Two classes of lone prepositions which are not followed by a complement are known as dangling and stranded prepositions. There are several contexts where these prepositions are found and they are very common in standard English. We will focus primarily on pseudo-passive and relative contexts, and mention another context in our conclusion.

One example of a pseudo-passive, that is a passive with a stranded preposition, is given in (1) and some relative clause examples are illustrated in (2), where the preposition is said to be dangling (Mel'čuk, 1987, p.124). The prepositions are indicated in bold in each case.

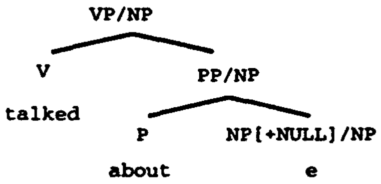
(1) Pseudo-passive (stranding)
He was yelled **at**.

(2) Relatives (with dangling prepositions)
a. They knew the man we talked **about**.
b. They knew the man who he thinks we talked **about**.

Note that these lone prepositions are not used in the same contexts as particles like *out* which forms a phrasal verb with *take* in the sentence *He took the garbage out*. In this latter case while there is a debate as to where the preposition should attach (to the verb *take* or to the NP *the garbage*), no NP is missing, or extraposed, contrary to the examples above.

Outside the realm of dependency grammar, in phrase structure grammar, the analysis of such sentences would have the lone preposition *about* belonging to the verb phrase headed by *talked*. In the phrase structure approach of Gazdar *et al* (1985, p.147), for example, there would be an empty category and SLASH notation, as indicated in (3).

(3) Generalized Phrase Structure Grammar analysis



In dependency grammar, the dependence relations are the crucial ones, rather than constituency. There have been different views on what relations a lone preposition bears to the other elements in these types of constructions. We will present analyses from the conceptions of dependency grammar proposed by Mel'čuk, and by Hudson, both of whom treat such constructions.

2.2 Mel'čuk's Analyses

According to Mel'čuk (1987, pp.82, 124-125), a preposition must have a dependent NP, except in the following cases. Stranded prepositions have no dependent, and dangling prepositions may or may not have the usual dependent. If the dangling preposition does have a dependent, it is not attached in the usual way, as we will illustrate.

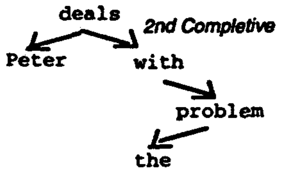
Starting with pseudo-passives, preposition stranding occurs when the dependent NP in an active construction becomes the grammatical subject in the related pseudo-passive construction. Here are two examples from Mel'čuk.

(4) Peter deals with the problem.
(5) The problem is dealt with by Peter.

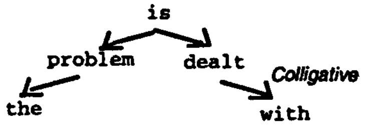
One consequence of passivization is the conversion of one of the surface syntactic relations, known as SSyntRels in Mel'čuk's terminology (see the discussion in Mel'čuk, 1987, p.31). In particular, the relation that subordinates the preposition (and its dependent NP) to the active verb in (4) is not the same as the relation between those elements in (5). The corresponding structures in (4a) and (5a) below are derived from the diagram in Mel'čuk (1987, p.124), with the passive agent omitted.

In (4a) the preposition *with* and its dependent *the problem* are subordinated to the verb *deals* by the *2nd Completive* SSyntRel. In contrast, in (5a) the latter SSyntRel is not tolerated in a passive construction and therefore a special SSyntRel, the *Colligative*, is posited especially for this construction.

(4a) Active



(5a) Pseudo-passive (stranding)



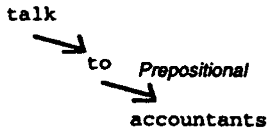
Mel'čuk makes a distinction between these stranded prepositions and dangling prepositions. The dangling preposition, unlike the stranded preposition, keeps the original SSyntRel that subordinates it to the verb.

Consider the data for dangling prepositions in relatives. The basic sentence in (6) has no relativization, while dangling prepositions can be found in sentences involving relative clauses such as (7) and (8) (examples from Mel'čuk) :

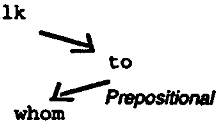
(6) I talked to all the accountants.
(7) All the accountants whom I talk to say receivables are piling up.
(8) All the accountants I talk to say receivables are piling up.

In (6) the normal *Prepositional* SSyntRel holds between the preposition and its complement, as illustrated in (6a) below. The dangling preposition in (7) continues to head a SSyntRel that subordinates its displaced complement *whom* labelled a *Prepositional* SSyntRel, as in (7a). Finally, in (8), since *whom* is deleted, there is no such relation and the preposition has no dependents, as in (8a) (structures adapted from Mel'čuk 1987, pp.130, 366):

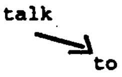
(6a) Preposition with usual complement



(7a) Dangling with relative pronoun



(8a) Dangling preposition without pronoun

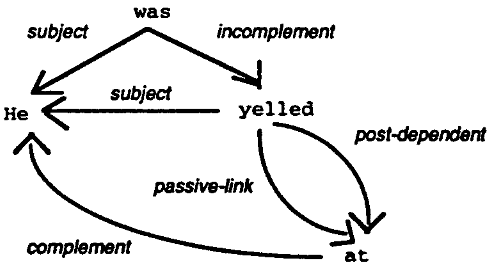


2.3 Hudson's Analyses

The analysis of lone prepositions in Hudson (1990) involves somewhat different relations. He points out that in his theory, Word Grammar, multiple relations between two elements are allowed, and that a word may depend on more than one head simultaneously (see the discussion in Hudson 1992, p.145).

Dangling prepositions in relatives are acknowledged but not given an explicit analysis in Hudson (1990). He does, however, provide the following analysis for a pseudo-passive with a stranded preposition (adapted from Hudson 1990, p.348).

(9) Stranded Preposition



Notice that the stranded preposition in (9) bears three dependency relations. It is a post-dependent of the verb *yelled*, it has the pronoun *he* as its complement, and it bears a relation special to pseudo-passives with stranded prepositions labelled *passive-link*.

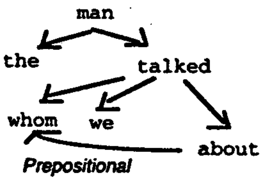
3 Implementation problems

There are a number of problems associated with the implementation of these theoretical approaches. Note that there is disagreement between linguists as to the optimal treatment. It is perhaps no coincidence that a problem which presents theoretical difficulties is also more problematic to implement.

In structures like (9) above, one node, namely the node containing the word *he*, has three fathers: *was*, *yelled* and *at*. We cannot implement this structure directly since in our implementation, each node in a structure has a unique father node and only adjacent nodes may be linked by a relation. This strategy reduces considerably the number of intermediate trees to be examined while constructing a given tree. It also simplifies traversal of trees.

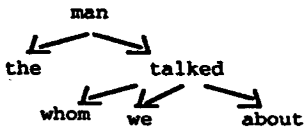
Consider next dangling prepositions in relatives, such as *the man whom we talked about*, which pose a similar problem. We cannot implement Mel'čuk's analysis illustrated in (10) since the node *whom* has two fathers: *talked* and *about*.

(10)



Our initial hypothesis for implementing such relatives was simply to attach the dangling preposition to the verb immediately to its left as in (11) while allowing other constraints to verify that the relative pronoun is correctly licensed within the structure.

(11)



This choice of implementation led to a number of serious efficiency problems. The main problem we will address is that too many trees were being produced and therefore too much time was being wasted.

A preposition must have a complement in order to attach to the verb with its normal relation. However, a dangling preposition attaches to the verb without its complement. In the course of analyzing every sentence containing a preposition after the verb, the preposition was attached both as dangling and not as dangling, since deciding whether a given preposition is dangling or not can be difficult locally. Even if the invalid analyses can eventually be discarded, their generation greatly increases parsing time. We will see in the next section how this problem disappears if the analysis is slightly modified.

Let us examine some other means of avoiding misanalysis and overgeneration of trees for these sentences, and show how these means are inadequate. First, one could verify the category of the word following the preposition since a lone preposition would not be followed by a nominal complement. This kind of restriction must be used with great care, especially in English in which words often belong to several categories and inflection is not rich enough to help disambiguate between categories. Consider some concrete examples such as (12) where the preposition *in* is not dangling and (13) where *about* is a dangling preposition. In (12), this restriction does not help the parser since *shops* can be a verb and yet it is the complement of the preposition. Similarly, in (13), checking the category of *shops* is not sufficient to determine that *about* is a dangling preposition since *shops* can be either a verb or a noun.

(12) He sold them in shops.
(13) The man we talked **about** shops here.

Secondly, one could propose a strategy where the preposition would attach to the verb only after the relative clause has been attached to the noun. This presupposes, however, that subtrees can be combined arbitrarily, i.e. by joining together any intermediate (non-root) nodes in the construction of the tree. This is problematic because it potentially creates trees with two roots. Moreover, this augmentation of the system is not warranted. We already have an efficient strategy that is not arbitrary which allows the combination of complete subtrees only.

4 Solutions

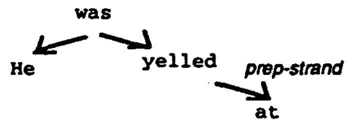
To avoid superfluous tree building, we allow lone prepositions to attach only at the point in the tree where they are licensed. That is, attachment of a lone preposition is permitted only when there is a context that permits such a preposition, such as a passive verb or a relative clause structure. We show how our solution reduced parsing time.

4.1 Pseudo-passives

For pseudo-passives, our implementation is closer to that of Mel'čuk than to that of Hudson. Recall that Hudson's analysis in (9) involved multiple fathers. We thus chose not to implement his *complement* relation between *at* and *he* and only a single relation between *yelled* and *at* in sentences like *He was yelled at.*

Following Hudson we use a distinctive relation, which we label *prep-strand* instead of *passive-link*. In addition, we use a set of constraints to check that the preposition is indeed appropriate to the verb (that *yell* can take *at*). Thus our analysis is as follows, where each node has at most one father and where only one relation holds between any pair of nodes:

(14) implementation of pseudo-passive



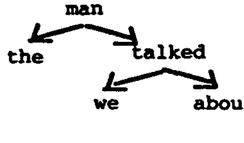
4.2 Relatives

While our solution for pseudo-passives closely follows that of Mel'čuk, the case of relatives is more complex. Recall that some of his analyses of relatives involved multiple fathers.

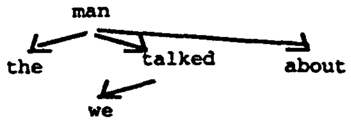
In order to avoid this problem in relative clauses, the dangling preposition is attached in our implementation not to the governing verb, but to the noun which is the antecedent of the relative. To see the advantage of our analysis, consider sentences where the relative is not introduced by a wh-word in examples like (15) (previously (2a)).

(15) The man we talked **about**.

(15a) our original solution



(15b) implemented solution

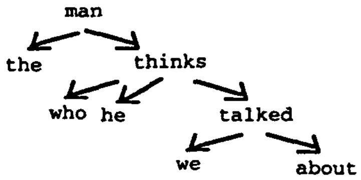


In this example, the dangling preposition is only licensed by the presence of the relative clause. Instead of (15a), we therefore prefer the analysis in (15b), where the preposition is attached to the head noun *man* once a relative clause has been created. Rather than implementing a relation between *talked* and *about* we verify compatibility between the verb and its prepositional complement independently. Note that an incomplete relative clause is created: *we talked*. Some constraints are relaxed and checked at a higher level to ensure the ultimate completeness of the overall structure.

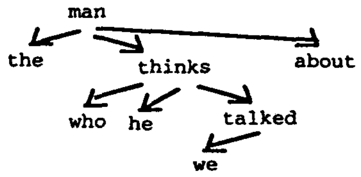
Next, consider long-distance relatives such as in (16) (previously (2a)) and the analyses in (16a) and (16b).

(16) The man who he thinks we talked **about**

(16a) our original solution



(16b) implemented solution



In (16), we combine the problem of dangling a preposition with that of unbounded dependency. Within our system, it is impossible to attach the relative pronoun *who* to the verb which subcategorizes for it in these long-distance relatives because of word order. In the same way, the dangling preposition *about* does not attach to *talked* but rather it attaches at a higher level, to *man*.

These analyses have crucially solved the problem of tree overgeneration. The attachment of lone prepositions may be made once the licensing criteria are met (passive voice, relativization or other such contexts). Therefore only those subtrees which will likely lead to a complete and successful analysis will be built.

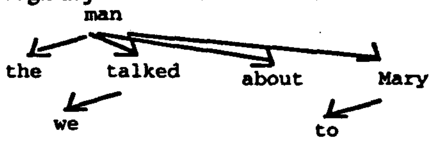
4.3 Remaining problems

Our analysis presupposes that the dangling preposition occurs as the last element in the relative clause. There are rare cases where another element can follow the dangling preposition, such as (17).

(17) The man we talked **about** to Mary

Since the dangling preposition *about* is attached to *man*, to avoid crossing of dependency relations, we would have to attach the phrase *to Mary* to the node *man* instead of attaching it more naturally to the verb *talked*. The analysis is shown in (18).

(18) tough-adjective with an extra PP



Note that there are cases where a prepositional phrase can attach to a noun following a relative. Thus the construction in (17) would have the same analysis as that in (19).

(19) The man we talked **about** with glasses

Constructions such as the one in (17) are not marked constructions. However, given their low frequency relative to the high frequency of preposition dangling in general, our constrained analysis is justified in terms of computational efficiency.

5 Conclusion

Our solution led to an overall improvement of the parser's performance. This type of solution is, of course, only one of many ways to reduce the size of the search space. We have found that the problem of combinatorial explosion in parsing English is even greater than it is in French due to the higher incidence of lexical ambiguity in English. Our adaptation of analyses found in the literature was therefore deemed necessary.

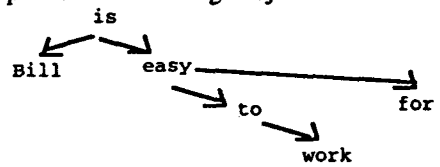
A related problem for which we have not come across a theoretical analysis is lone prepositions in the context of so-called tough-adjectives. These adjectives can take an infinitival complement whose object is missing (as in (20)). Infinitives with a prepositional object are also possible complements of tough-adjectives (as in (21)), and this is another context where a lone preposition is licensed, as exemple (22) illustrates.

(20) Bill is easy to love.
(21) It is easy to work for Bill.
(22) Bill is easy to work **for**.

Sentences like (22) require a complex theoretical analysis. A dependency relation should hold between the preposition *for* and the noun *Bill* while the latter is also the subject of the tough-adjective predicate.

Our analysis for this case was dictated by the same considerations as for the other cases. While the preposition depends on the preceding verb, it is licensed by the presence of the tough-adjective. Just as with the other cases, then, the preposition is attached high up in the structure at the point where it is licensed. Here, *for* is attached to the adjective, after the infinitival complement has been attached.

(23) implementation of tough-adjective construction



Our solution to the problem of lone prepositions has been influenced primarily by considerations of implementation. It remains to be seen what types of consequences this adaptation entails in terms of semantics.

In conclusion, we have presented a set of data that highlights an important constraint on many implementations, including our own: Linguistic descriptions must be modelled in such a way as to optimize performance.

Acknowledgements

We would like to thank *Les Logiciels Machina Sapiens inc.* for supporting us in writing this paper. We are endebted to all the people, past and present, who have contributed to the development of the grammar checkers.

We thank Mary Howatt for editing advice and anonymous reviewers for their useful comments. All errors remain those of the authors.

References

Fuchs, Catherine, Laurence Danlos, Anne Lacheret-Dujour, Daniel Luzzati and Bernard Victorri (1993) *Linguistique et traitements automatiques des langues*, Hachette, Paris.

Gazdar, Gerald, Ewan Klein, Geoffrey Pullum and Ivan Sag. (1985).Generalized Phrase Structure Grammar, Harvard University Press, Cambridge.

Hudson, Richard (1984) *Word Grammar*, Basil Blackwell, Oxford.

Hudson, Richard (1990) *English Word Grammar*, Basil Blackwell, Oxford.

Hudson, Richard and Norman Fraser (1992) "Inheritance in Word Grammar", in *Computational Linguistics* 18.2, MIT Press.

Mel'čuk, Igor A. (1987) *Surface Syntax of English. A Formal Model within the Meaning-Text Framework*, Benjamins, Amsterdam.

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