Segmentation of Complex Sentences*

Vladislav Kuboň[1], Markéta Lopatková[1], Martin Plátek[2], and Patrice Pognan[3]

[1] ÚFAL MFF UK, Prague, {lopatkova,vk}@ufal.mff.cuni.cz
[2] KTIML MFF UK, Prague, martin.platek@mff.cuni.cz
[3] CERTAL INALCO, Paris, mcertal@wanadoo.fr

Abstract. The paper describes a method of dividing complex sentences into segments, easily detectable and linguistically motivated units that may be subsequently combined into clauses and thus provide a structure of a complex sentence with regard to the mutual relationship of individual clauses. The method has been developed for Czech as a language representing languages with relatively high degree of word-order freedom. The paper introduces important terms, describes a segmentation chart, the data structure used for the description of mutual relationship between individual segments and separators. It also contains a simple set of rules applied for the segmentation of a small set of Czech sentences. The segmentation results are evaluated against a small hand-annotated corpus of Czech complex sentences.

1 Introduction

It is quite obvious that the syntactic analysis of long and complicated natural language sentences is more difficult than the analysis of short sentences. A parsing success depends among other things also on the length of the input sentence. This has been shown very often in the past, let us mention for example [1], [2] for rule-based syntactic analyzers and [3] for stochastic parsing of Czech.

There are also multiple solutions to the problem of bridging the gap between results of morphological analysis (or tagging) and a full-scale rule-based syntactic analysis or stochastic parsing. Let us mention for example the idea of cascaded parsing used in [4], [5] or [6]. The advantage of working with a cascade of specialized parsers instead of having one very complex general parser is quite obvious – the complexity of the task is substantially reduced and the parsing process is speeded up.

The use of chunking[4] is also quite frequent. The identification of chunks prior to parsing helps to decrease the parsing complexity, the only problem being the correct identification of chunks – if it is done only on the basis of very limited local context (bigrams or trigrams), it may be misleading with regard to the context of the whole sentence.

* This paper is a result of the project supported by the grant No. 1ET100300517.

[4] Very comprehensive explanation of this notion can be found for example at http://nltk.sourceforge.net/tutorial/chunking/

Very interesting approach to dividing the parsing process into several relatively independent but mutually closely related parts has been introduced in the XDG theory of D. Duchier and others, see [7]. We think that the idea presented in this paper may be exploited especially in connection with similar approaches.

This paper describes a method how to estimate the structure of clauses (their span and mutual relationships) solely on the basis of results of morphological analysis of an input sentence and very strict syntactic rules concerning punctuation.

Although the method presented in this paper had been designed for the syntactic analyzers of Czech, it is rather useful for a whole group of related and typologically similar languages. Some papers (e.g. [8]) indicate that the punctuation is important even for languages of a different type. It is not true that the information allowing to divide the complex sentence into individual clauses or segments is not important and that every stochastic parser will provide it for free in the parsing process – the substantially lower results (almost 10% difference) reported for Czech compared to English for identical parsers (see [3], [9]) support the claim that even stochastic parsers have difficulties to cope with free-word order languages.

2 Describing a structure of a complex sentence

The basic idea underlying our method is an assumption that every morphologically analyzed sentence already contains a lot of more or less reliable information that may be directly used for the benefit of more effective and precise syntactic parsing. We exploit Czech grammars (esp. [10]) as well as previous linguistic observations (see [11]).

The most important information we are looking for is the information about the mutual relationships between individual clauses, the span of embedded clauses etc. Let us call this type of structural information a **clause structure** of the (complex) sentence. At the beginning it is important to stress that we suppose neither that our method will be able to provide an unambiguous clause structure for every sentence nor that an unambiguous clause structure exists for every sentence. The aim is to create as precise an approximation of the clause structure as possible.

2.1 Important notions

In the sequel an input sentence is understood to be a sequence of lexical items $w_1 w_2 \dots w_n$. Each item w_i $(1 \leq i \leq n)$ represents either a certain lexical form of a given natural language, or a punctuation mark, quotation mark, parenthesis, dash, colon, semicolon or any other special symbol which may appear in the written form of a sentence. All items are disjunctively divided into two groups – ordinary words and separators.

Let us call the words or punctuation marks which may separate two clauses (or two sentence members) **separators**. It is quite clear that there are at least three relatively easily distinguishable types of separators – opening ones, closing ones and mixed ones, those, which typically close the preceding clause or its part and open the following one. A typical opening separator is e.g a subordinating conjunction or a relative pronoun, a closing one is a full stop, question mark or exclamation mark at the end of a sentence, mixed separators are for example commas or coordinating conjunctions.

It is often the case that two clauses are separated by more than one separator (e.g. comma followed by *že* [that]), in some cases even combined with non-separators (emphasizing adverbs, prepositions, etc.). In such a case it would be more convenient to consider the whole sequence as a single item – let us call it a **compound separator**.

Let $S = w_1 w_2 \ldots w_n$ be a sentence of a natural language. A **segmentation of a sentence** S is a sequence of sections $D_0 W_1 D_1 \ldots W_k D_k$, where particular section W_i $(1 \leq i \leq k)$ represents so called **segment**, i.e. a (maximal) sequence of lexical items $w_j w_{j+1} \ldots w_{j+m}$ not containing any separator, and section D_i $(0 \leq i \leq k)$ represents a (compound) separator composed of items $w_q w_{q+1} \ldots w_{q+p}$. The section D_0 may be empty, all other sections D_i $(1 \leq i \leq k)$ are non-empty. Each item w_i for $1 \leq i \leq n$ belongs to exactly one section D_j if it is a member of a (compound) separator; in the opposite case, w_i belongs to exactly one W_j. A pair $D_{i-1} W_i$ (where D_{i-1} is an opening or mixed (compound) separator) is called an **extended segment**.

The section D_0 is usually empty for sentences which start with a main clause. D_0 is typically nonempty if a complex sentence starts with a subordinated clause, as e.g. in the sentence *Když jsem se probudil, zavolal jsem policii.* [When I woke up, I called the police.]. D_k represents the final punctuation mark at the end of a sentence.

The segmentation of a particular sentence can be represented by one or more **segmentation charts** that describes the mutual relationship of individual sections with regard to their coordination or subordination.

Each separator is represented by at least one node. If an opening separator represented by a node D_i has a subordinating function, a copy of the node D'_i is placed directly under a node D_i in the chart and it is connected by a dotted arrow with the original node D_i. The closing separator may by also represented by a "raised" copy of a node D_i. Let us demonstrate example of a segmentation chart on the Czech complex sentence *Zatímco neúspěch bývá sirotkem, úspěch mívá mnoho tatínků, horlivě se hlásících, že zrovna oni byli u jeho početí.* [While failure is usually an orphan, the success tends to have many fathers, claiming eagerly that particularly they were present at its conception.], see Fig. 1.

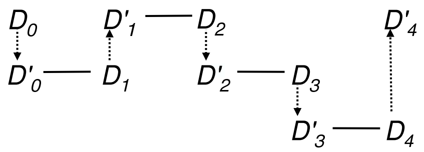

Fig. 1. Example of segmentation chart

D_0 - *Zatímco* [While]
W_1 - *neúspěch bývá sirotkem* [failure is usually an orphan]
D_1 - *,*
W_2 - *úspěch mívá mnoho tatínků* [the success tends to have many fathers]
D_2 - *,*
W_3 - *horlivě se hlásících* [claiming eagerly]
D_3 - *, že* [that]
W_4 - *zrovna oni byli u jeho početí* [that particularly they were present at its conception]
D_4 - *.*

There is more than one chart in case that the segmentation of a sentence is ambiguous. It may happen if a separator is ambiguous – e.g. the Czech word form *jak*, which may be both a noun or a subordinating conjunction – or if a separator does not clearly indicate the relationship between both segments it separates, as e.g. comma.

In order to be able to present a basic set od rules for creating segmentation chart it is necessary to introduce a couple of new notions, at least informally.

A **subordination flag** is assigned to particular extended segment either if this segment contains any word form with one of the following morphological tags (for conjunctions, pronouns, and numerals, see [12]) or if it contains one of the listed pronominal adverbs:

- tag="J,.*" representing a subordinating conjunction;
- tag="P.*" representing a interrogative/relative pronoun, where the second position in the tag contains any of the following characters:
  - 4 (*jaký, který, čí, ...*),
  - E (*což*),
  - J (*jenž, již, ...*),
  - K (*kdo, kdož, kdožs*),
  - Q (*co, copak, cožpak*),
  - Y (*oč, nač, zač*);
- tag="C.*" representing numerals, where the second position in the tag is either
  - ? (*kolik*),
  - u (*kolikrát*) or
  - z (*kolikátý*);
- tag="D.*" for pronominal adverbs
  - adverbs (jak, kam, kde, kdy, pro)

For the sake of an easier explanation of mutual relationships of individual nodes of a segmentation chart in vertical direction we would like to introduce the notion of **chart layers**. In informal terms, a top layer of the chart (layer 1) corresponds to a main clause of the sentence and the numbers identifying layers increase in the top-down direction. The lower layers (layers with higher numbers) represent subordinated clauses. If a clause contains an embedded clause (fully embedded, that is the main clause is divided into two non-empty parts), the "tail" of the main clause is located in the same layer as its "head"; the same holds also for subordinated clauses with more deeply embedded clauses.

2.2 General principles of building segmentation charts

The process of building segmentation charts is relatively straightforward. In accordance with the principles presented above, the first step is always the morphological analysis of the input sentence. On the basis of its (typically ambiguous) results we will divide the sentence into segments, taking into account the number and position of all separators and (compound) separators in the sentence.

The next step, drawing segmentation charts relevant for a given input sentence, is slightly more complicated due to the ambiguity concerning especially closing separators (mainly commas), which are generally highly ambiguous. Not only they can simply raise, lower or directly connect the following section at the same layer, they may even raise the following section several layers (in case of closing a deeply embedded subordinated clause). If there is such an ambiguous separator anywhere in the sentence, it is necessary to create more segmentation charts, each with an edge going in a different direction.

2.3 Basic set of rules

In order to demonstrate how the process of building the segmentation chart works, we present here a basic set of rules for Czech:

1. **Sentence start:** If the first (extended) segment does not have a subordination flag the edge representing the first segment starts at the topmost (1^{st}) layer of the chart and continues straight to the right. Otherwise the edge for first segment starts at the 2^{nd} layer.
2. **Comma:** If the comma is NOT followed by an item with a subordination flag, the next segment goes either straight to the right (this represents for example a comma separating two coordinated items inside a single clause) OR it jumps one or more layers (this is a highly ambiguous situation representing an end of an nested subordinated clause) upwards.
3. **Comma followed by an item with a subordination flag:** In this case the next segment moves downward.[5]
4. **Coordinating expression:** Coordinating conjunction or any other coordinating expression preserves a layer, even though it might be followed by an extended segment with subordination flag.
5. **Full stop, question mark, exclamation mark:** These characters represent an end of the sentence, therefore the last node of the segmentation chart always jumps to the 1^{st} layer of the chart (the layer of the main clause).
6. **Opening quotation marks:** Opening quotation marks are considered to be a separator only when they are at the start of the sentence or when they

[5] There are some exceptions to this general rule, which may be handled by a set of conditions capturing those specific constructions allowing to go either right or to move the next segment upwards. Such a construction may be found for example in the sentence *Řekl, že byl, jaký byl, ŽE je, jaký je a že bude, jaký bude.* [(He) said that (he) was who (he) was, that (he) is who (he) is and the (he) is going to be who (he) is going to be.]

are combined with other separators (comma, semicolon etc.) – in such a case the next segment jumps one layer down.

7. **Closing quotation marks:** They are a separator only if they follow opening quotation marks, which are considered being a separator as well – in such a case the next segment jumps one or more layer up.

3 Evaluation

The evaluation of our method turned out to be more complicated than we have originally envisaged. We have assumed that the richly syntactically annotated Prague Dependency Treebank[6] will provide large enough set of sentences, but it turned out that this assumption has been wrong.

The problem is the annotation – there are too many syntactic phenomena for which it is extremely difficult, if not impossible, to find a general consensus about annotation. A huge number of decisions has to be made concerning the annotation of complex linguistic phenomena like coordination, verbal complexes, the proper place of prepositions etc.

This inevitably leads to difficulties when someone tries to search the corpus for an information which had not been accounted for at the moment of the annotation scheme design. Let us demonstrate this on a very simple example – nothing is probably more easy to determine as a single unit than a pair of parenthesis inside a sentence. Unlike punctuation signs, the parenthesis unambiguously show the beginning and the end of a text inserted into clause. It is therefore quite natural to expect this easily detectable segment to be annotated in one way.

It turned out that this is not the case of the analytical level of PDT. After an examination of a small sample of the treebank we have found as many as 7 different ways how the parenthesis (and their content) were annotated in a certain context. Let us show at least two of those cases, both even located in the same sentence (see particular subtrees in Fig. 2): *Před několika dny vypukl další skandál (privatizace Čokoládoven v Modřanech), v němž byl do role hlavního viníka opět obsazen Fond národního majetku (FNM) a jeho předseda Tomáš Ježek.* [Yet another scandal erupted few days ago (a privatization of Čokoládovny in Modřany), in which the main role was played by a National Property Fund (NPF) and its chairman Tomáš Ježek.]

Not only the annotation of a content of both parenthesis differs, but even the mutual position of both types of parenthesis in the tree is different. It is quite clear that the transformation of sentences from PDT would require a lot of manual effort in order to provide a good testing material for our method.

These considerations led us to a decision to annotate manually a small sample of text not according to the standard of PDT, but according to the definition of the segmentation chart. Two articles from a daily newspapers Lidové noviny and

[6] http://ufal.mff.cuni.cz/pdt2.0/

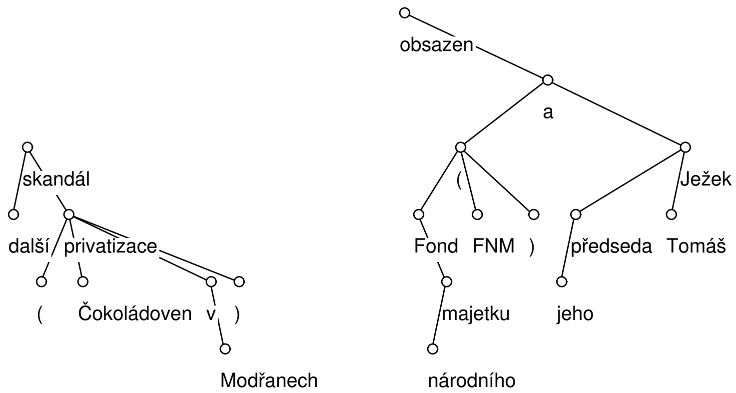

Fig. 2. Two types of parenthesis annotation in PDT

Neviditelný pes[7] (LN, resp. NP in Table 1) containing political commentaries have been selected and manually annotated as a test set.

The table below shows the degree of ambiguity of segmentation charts created automatically using the set of rules presented above, i.e. very local rules which do not presuppose understanding the sentence meaning.

Table 1. Degree of ambiguity of segmentation charts

	number of			number of charts					
	sentences	tokens	segments	1	2	3	4	5	more
LN	33	553	78	28	2	1	1	1	-
NP	15	334	57	12	3	-	-	-	-
total	48	887	135	40	5	1	1	1	-

Even though the test set is relatively small, the table clearly shows that the simple rules presented above provide a very good starting point and that in the average case the segmentation charts are almost unambiguous when a real text is concerned. It is of course possible to find very elaborated examples of sentences where our simple rules fail produce high number of segmentation charts, but the further refinement of those simple rules may improve even that. The most important result of the test was the 100% coverage of our method – not a single correct segmentation chart has been omitted by our algorithm.

7 http://pes.eunet.cz

4 Conclusion

The method presented in this paper shows that (at least for a language displaying inflectional morphology similar to that of Czech) it is possible to draw a chart reflecting the mutual position of clauses or their parts (segments) in complex sentences without applying the full-fledged syntactic parsing of the whole sentence first. The method is based on the identification of separators and their classification. The subsequent steps in the parsing process (which are not covered by this paper) may then decide, on one hand, which of the charts is not valid (in case that there are several variants of charts as an output of our method), and, on the other hand, exploit the charts for faster and more effective syntactic analysis of complex sentences. The evaluation of the method presented in the paper indicates that the segmentation may really help, the ambiguous segmentation charts are more or less rare.

The results achieved so far encourage further research in two areas. The first area concerns further development of more precise segmentation rules, the second one might concern the step from segmentation charts towards the chart reflecting the mutual position of clauses, not only segments.

References

1. Oliva, K.: A Parser for Czech Implemented in Systems Q. In: Explizite Beschreibung der Sprache und automatische Textbearbeitung, MFF UK Praha (1989)
2. Kuboň, V.: Problems of Robust Parsing of Czech. Ph.D. Thesis, MFF UK, Prague (2001)
3. Zeman, D.: Parsing with a Statistical Dependency Model. Ph.D. Thesis. MFF UK, Prague (2004)
4. Abney, S: Partial Parsing via Finite-State Cascades. In: Journal of Natural Language Engineering, Vol 2, No 4 (1995) 337–344
5. Ciravegna, F., Lavelli, A.: Full Text Parsing using Cascades of Rules: An Information Extraction Procedure. In: Proceedings of EACL'99, University of Bergen (1999)
6. Brants, T.: Cascaded Markov Models. In: Proceedings of EACL'99, University of Bergen (1999)
7. Debusmann, R., Duchier, D., Rossberg, A.: Modular grammar design with typed parametric principles. In: Proceedings of FG-MOL 2005, Edinburgh (2005)
8. Jones, B.E.M.: Exploiting the Role of Punctuation in Parsing Natural Text, In: Proceedings of the COLING'94, Kyoto, University of Kyoto (1994) 421–425
9. Hajič, J., Vidová-Hladká, B., Zeman, D.: Core Natural Language Processing Technology Applicable to Multiple Languages. The Workshop 98 Final Report. Center for Language and Speech Processing, Johns Hopkins University, Baltimore (1998)
10. Šmilauer, V.: Učebnice větného rozboru. SPN, Praha (1958)
11. Holan, T., Kuboň, V., Oliva, K., Plátek, M.: On Complexity of Word Order. In: Les grammaires de dpendance – Traitement automatique des langues, Vol 41, No 1 (2000) 273–300
12. Hajič, J.: Disambiguation of Rich Inflection (Computational Morphology of Czech). UK, Nakladatelstv Karolinum, Praha (2004)