Obtaining Hidden Relations from a Syntactically Annotated Corpus - From Word Relationships to Clause Relationships

Oldřich Krůza and **Vladislav Kuboň**
Faculty of Mathematics and Physics
Charles University in Prague
{kruza,vk}@ufal.mff.cuni.cz

Abstract

The paper concentrates on obtaining hidden relationships among individual clauses of complex sentences from the Prague Dependency Treebank. The treebank contains only an information about mutual relationships among individual tokens (words, punctuation marks), not about more complex units (clauses). For the experiments with clauses and their parts (segments) it was therefore necessary to develop an automatic method transforming the original annotation into a scheme describing the syntactic relationships between clauses. The task was complicated by a certain degree of inconsistency in original annotation with regard to clauses and their structure. The paper describes the algorithm of deriving clause-related information from the existing annotation and its evaluation.

Introduction

One of the major factors which changed linguistics during the past twenty years was without a doubt a strong stress on building and exploiting large annotated corpora of natural languages. They serve nowadays as a primary source of evidence for the development and evaluation of linguistic theories and applications.

Although the corpora are extremely important source of data, they are not omnipotent. The more elaborated annotation scheme the authors use, the more problems with linguistic phenomena they have to solve. It is relatively easy to annotate even very large corpus with simple part-of-speech annotation if the natural language being annotated has relatively limited inflection (and thus also its morphological variation is relatively limited). It is much more difficult to create a consistently annotated treebank. Such an annotation requires making a large number of decisions about a particular annotation of particular linguistic phenomena. It is natural that not all phenomena are taken into account, it is also natural that some of the phenomena taken into considerations sometimes collide (e.g. when a particular word is affected by more than one phenomenon, each of which requires a different style of annotation). The more elaborated and detailed is the annotation, the easier it is to find phenomena which are annotated in a seemingly inconsistent way. If the annotation is really well-designed and consistent then it should be possible to extract an information hidden in the corpus or treebank even in case that a particular phenomenon we are interested in was not annotated explicitly.

This paper describes an attempt to do precisely that - to extract an information which may be useful for research of a particular linguistic phenomenon from the treebank, where this phenomenon is not explicitly tagged. The treebank under consideration is the Prague Dependency Treebank (PDT)[1], a large and elaborated corpus with rich syntactic annotation of Czech sentences. The phenomenon we are interested in are Czech complex sentences, the mutual relationship of their clauses and properties of those clauses. In the following sections we would like to present a brief description of the PDT, followed by a discussion of the annotation of clauses in complex sentences (and their parts - segments) in the PDT. Then we are going to describe an automatic method how to extract the required information from PDT (where it is not explicitly marked). In the last section we are going to present a discussion concerning the methods and results of an evaluation of the method presented in the paper.

The Prague Dependency Treebank

The Prague Dependency Treebank is a result of a large scale project started in 1996 at the Faculty of Mathematics and Physics at the Charles University in Prague . It is a corpus annotated on multiple levels - morphological, analytical and underlying-syntactic layer (for a description of the tagging scheme of PDT, see e.g. Hajič 1998, Hajič and Hladká 1997, Hajičová 1998, 1999, and the two manuals for tagging published as Technical Reports by UFAL and CKL of the Faculty of Mathematics and Physics, Charles University Prague (see Hajič et al. 2001) and available also on the website http://ufal.mff.cuni.cz). The annotation on the underlying syntactic level the result of which are the so-called tectogrammatical tree structures is based on the original theoretical framework of Functional Generative Description as proposed by Petr Sgall in the late sixties (see Sgal et al. 1986) and developed since then by the members of his research team.

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

Problems with Clauses in the PDT

Unfortunately, although the annotation scheme of PDT allows for a very deep description of many kinds of syntactic relationships, there is no explicit annotation of the mutual relationships of individual clauses in complex sentences in the corpus.

A sentence at the analytical layer is represented as a dependency tree, i.e. a connected acyclic directed graph in which no more than one edge leads upwards from a node. The nodes – labeled with complex symbols (sets of attributes) – represent individual tokens (wordforms and punctuation marks); one token of the sentence is represented by exactly one node of the tree. The edges represent syntactic relations in the sentence. The actual type of the relation is given as a function label of the edge, so called analytical function. In addition, linear ordering of the nodes corresponds to the original sentence word order.

In particular, there are no nonterminal nodes in PDT that would represent more complex sentence units – such units are expressed as (dependency) subtrees. This rule is applied generally - even the relationships where having a node representing a complex unit (such as coordination or complex verb form) would benefit the simplicity of the representation, are not marked by any nonterminal or artifficial node. It is therefore no wonder that also the clauses and their mutual relationship are not marked explicitly in the tree.

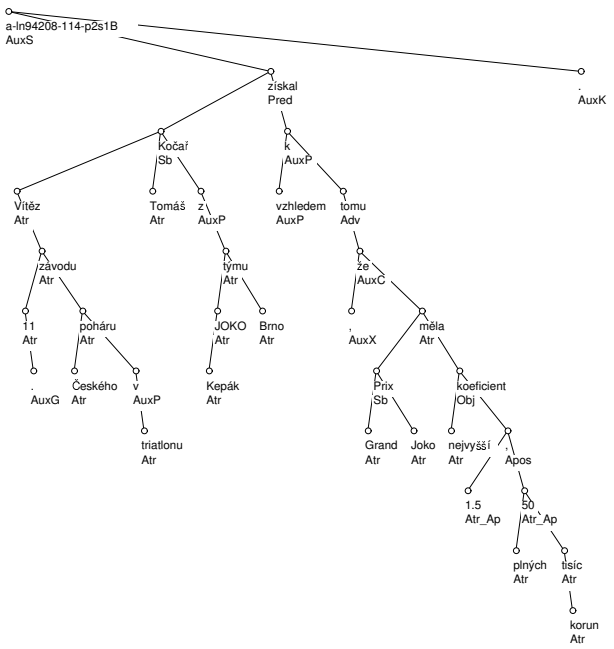

Figure 1: Analytical tree of the sentence "Vítěz 11.závodu Českého poháru v triatlonu Tomáš Kočař z Kepák JOKO týmu Brno získal vzhledem k tomu, že Grand Prix Joko měla nejvyšší koeficient 1.5, plných 50 tisíc korun." (The winner of the 11th race of the Czech Triathlon Cup Tomáš Kočař from Kepák JOKO team Brno obtained the full amount of 50 thousand thanks to the fact that the Joko Grand Prix had the highest coefficient of 1.5.)

Let us demonstrate the problem of reconstructing the clauses from the analytical level of the PDT on a sample complex sentence from the PDT (Figure 1). The complex sentence has a main clause which is divided into two parts by a inserted subordinated clause , *že Grand Prix Joko měla nejvyšší koeficient 1.5*, (that Joko Grand Prix had the highest coefficient of 1.5). Although the separated tail of the main clause contains a direct object *50 tisíc korun* (50 thousand crowns) of the verb *získal* (he obtained), their relationship in the tree is far from direct. In order to retrieve the mutual relationship of all three sections (representing the two clauses of the sentence), the algorithm has to dig in very deep into the tree and it also must be able to recognize that the subtree rooted in the node *50* is actually not a subtree of the subordinated clause, but a subtree of the governing verb of the whole sentence.

Clauses from the Viewpoint of Surface Syntax

The definition of clauses in this paper is based on the analytic level of PDT. Out of the three main levels of annotation of the PDT (morphemic, analytic, tectogrammatic), this level is the only one which describes surface syntax of Czech sentences. It is therefore the most suitable level for capturing the mutual relationships of clauses in complex sentences. By taking advantage of the analytic annotation, we could come up with a simple definition that only minimally refers to language intuition and meaning.

How to Identify a Clause

A clause is defined as a subtree of a predicate, including the predicate, with the exception that 1) a subordinating conjunction governing the predicate belongs to the clause and 2) a clause whose predicate is in the subtree of another clause is not considered to be a part of the governing clause.

Since not every predicate is explicitly annotated as such in the analytic level of PDT, this amounts to

1. tokens explicitly marked as predicates (those with analytic function "Pred"),
2. finite autosemantic verbs,
3. tokens that govern a node with the analytic function "AuxV"[2] and
4. tokens that are coordinated with a predicate (recursion occurs here).

Some special cases apply:

- ad 2: Finite verbs that hold coordination or apposition are not considered to be predicates for our purposes. See subsection .
- ad 3: If the token governing an AuxV-node is a coordinating conjunction, then it is not considered a predicate. In such a case though, the coordinated tokens are considered as if they governed an AuxV-node and are thus recognized as predicates.

[2]This denotes predicates formed by compound verbs

Relations among Clauses

Sentence sections The criteria introduced above state what is a clause and what tokens belong to one. This is one of the two goals of our algorithm. The second one deals with relations between and among clauses. These are basically dependency and coordination relations. Since clauses are not atomic objects and there are cases where a token belongs to more than one clause, we need to introduce a new, more general term: *sections*. The reason why we cannot use the notion of segment instead of a section is the different nature of both components of a clause. Segments are units distinguishable on the surface, they are defined for sentences in the form of sequences of word forms and punctuation marks. Sections, on the other hand, are defined as units distinguishable from the surface syntactic (analytic) representation of a sentence. Both terms refer to units which are similar but not identical.

A section of a sentence is defined by its *representative* and its *component*. The representative of a section is a token of the sentence or its technical root (every sentence in PDT has a technical root). Each token as well as the technical root represents no more than one section. The component of a section is a subset of the tokens of the sentence. The components of the sentence's sections constitute a perfect coverage of the sentence's tokens.

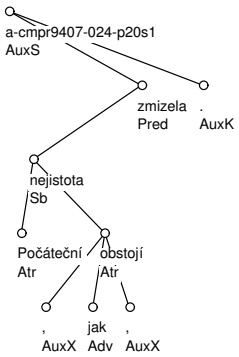

Figure 2: Analytic tree of the sentence "Počáteční nejistota, jak obstojí, zmizela." [Initial uncertainty, how it-will-do, vanished.]

Typically, the representative of a section belongs also to its component. The exception is the technical root, which can represent a section but can never be in its component.

The component of a section forms a tree on the analytic level. The only exception is the section represented by the technical root, the component of which can be a forest.

Sections are of three types:

1. clause,
2. coordination and
3. adjunct.

Each clause constitutes a section of the type *clause*. Its representative is the finite verb that governs the clause and its component consists of the tokens that belong to the clause.

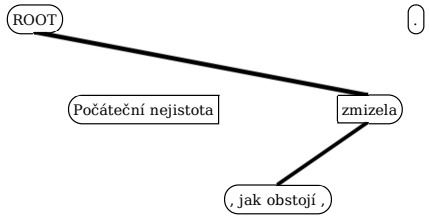

Figure 3: Sections of the sentence from Figure 2. Each bordered shape marks a section. Each horizontal level contains one clause. The lines mark bloodline relations of clauses.

Sections to Describe Coordination Whenever two or more clauses are coordinated, the coordination itself constitutes an extra section of the type *coordination*. Its representative is the coordinating conjunction (or punctuation token) that holds the coordination (i.e. it has the analytic function of "Coord" or "Apos" in case of appositions, which we treat equally to coordinations). The component of a coordination section is its representative and leaf tokens dependent on it that are not related to the coordinated clauses. That is:

1. other conjunctions, commas and other separators of the coordinated clauses,
2. other words of the conjunction in case of multi-word conjunctions,
3. auxiliary leaf tokens (those with analytic function beginning with "Aux").

The third case emerges when a coordination of clauses governs a phrase that effectively depends on all the coordinated clauses. Take the English example: *"John loves Mary but won't marry."* There are two finite verbs present: "loves" and "won't". So our above stated definition would recognize two clauses plus one coordination.

Clause 1 would certainly contain tokens "loves Mary", Clause 2 would contain "won't marry" and the coordination would only contain "but". So, where would "John" go? It's him who loves Mary and it's also him who won't marry the poor girl. We could see this sentence as a coordination of clauses with the subject distributed: "John loves Mary" + "John won't marry".

To denote this type of relation, we give "John" (with his whole (empty) subtree) his own section of the type *adjunct*. Sections of this type are always formed by subtrees (dependent clauses excluding) of tokens that are not clauses and depend on a coordination of clauses but are not coordinated in it.

Tokens that do not fall into any section by the above criteria belong to the section represented by the technical root. Its type is set to *clause*, but that is a purely technical decision.

Relations Formally The sections were defined with the intention to obtain a help when capturing relations among clauses. Notice that every section's component has a tree-

like structure. The only exception again being the clause whose representative is the technical root. This means that every section has one root token. We can therefore define bloodline relations between sections like this:

Definition 1 (Parent section) Let D be a section whose representative is not the technical root. Let r be the root token of D. Let p be the analytic parent token of r. We call the section to which p belongs or which p represents the *parent* section of D. The root section is its own parent.

As in the real life, one child is sometimes quite unlike another, that is an experience of many human parents. It's the same here, so we differentiate several *types of children*. These are:

1. dependants,
2. members and
3. parts.

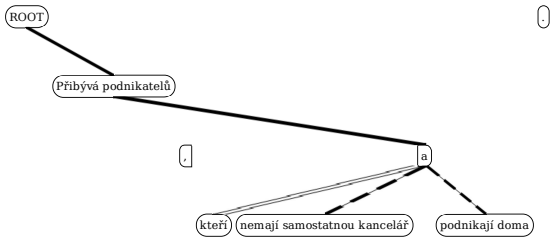

Figure 4: Sections and their relations of the sentence "Přibývá podnikatelů, kteří nemají samostatnou kancelář a podnikají doma." [The-number-grows of-businessmen, who don't-have separate office and work at-home.] (The number of businessmen who have no separate office and work at home grows.) The main clause "Přibývá podnikatelů" (the number of businessmen grows) governs the coordination formed by the conjunction and the comma. There are two dependent clauses: "kteří nemají samostatnou kancelář" (who have no separate office) and "kteří podnikají doma" (who work at home). Their disjoint parts are marked as coordinated sections (*section type: clause, child type: member*) and their common word "kteří" (who) is marked as another section (*section type: adjunct, child type: part*).

Every clause and every adjunct can only have child sections of the *dependant* type. Coordinations, on the other hand, can have children of any type.
Whenever a coordination has a child of the *member* type, it means that the child section is coordinated in the coordination.
Whenever a coordination has a child of the *dependant* type, it means that the child section is effectively dependent on all the sections coordinated in the parent coordination.
Whenever a coordination has a child of the *part* type, it means that the child section belongs to all the sections coordinated in the parent coordination. Children of the *part* type are exactly the sections of the type *adjunct*.

This approach allows to capture virtually any clause structure from the PDT, keeping information about tokens belonging to clauses, their dependencies and coordinations. The grammatical roles of clauses are easily extracted from analytic functions of their representatives.

Since the definitions mentioned above are all based on information available on the analytic and lower levels of PDT, the algorithm for extracting clause structure from the analytic annotation is a straight-forward rewrite of those definitions into a programming language.

Verbs Acting as Conjunctions

The only phenomenon we know that our algorithm is not handling correctly concerns finite verbs that bear an apposition (or potentially coordination), that is, they have the analytic function of "Coord" or "Apos". Take the sentence "Do úplných detailů jako jsou typy obkladaček nelze jít." [Into sheer details like are types of-tiles is-not-possible to-go.] (We can't go into sheer details like the types of tiles.) Figure 5 shows its analytic tree. The clause that should apparently be recognized is formed by tokens "jako jsou typy obkladaček" [like are types of-tiles]. Notice that the tokens "úplných detailů" [sheer details], which do not belong to the inner clause, are in the subtree of the inner clause's founding verb "jsou" [are].

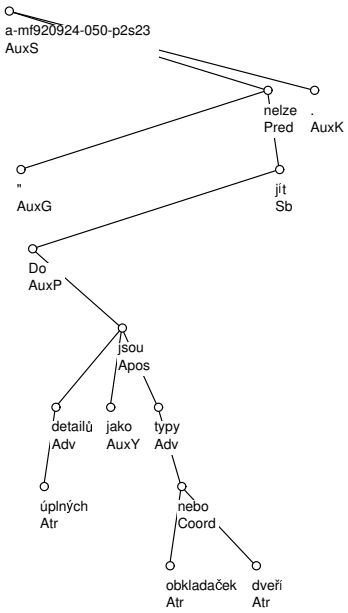

Figure 5: Analytic tree of a sentence containing an apposition held by a finite verb

Here, the most profound rule our definition is based upon – that a clause is a subtree of its predicate – breaks the factual distribution of clauses. Even if we only tore off the clause itself (which is a well-formed tree), the parent clause would stop being connected. Our way of dealing with this is to simply ignore the presence of the inner clause and keep it as a part of the parent clause. This seems to be the best way

to go, as it has virtually no negative consequences, it is very easy to detect and implement, and the phenomenon is not frequent.

Evaluation

Applying the algorithm described above on the PDT, we get clauses marked up in the sentences. The process is deterministic, it reflects the annotation of individual nodes of an analytic tree of the PDT. It is also an application of a *definition*, not an attempt to model a given linguistic phenomenon. The data should then be used as gold standard for clause detection from the lower levels of annotation (like morphological). It is clear that standard precision/recall evaluation would not tell us anything in this case.

What we decided to do instead is to try to count the sentences where the algorithm provides clauses in a different manner than we think a human would. The difference between automatic and man-made annotation is based upon the fact that our algorithm keeps clauses syntactically compact, while humans prefer to keep them linearly compact. These requirements go against each other mostly in the case where a coordination section has tokens inside some of the coordinated clauses. See Figure 6. Other cases include erroneously annotated trees in the corpus (garbage in – garbage out) and the presence of adjunct segments, which humans tend to connect to the adjacent clause only.

Table 1 presents the evaluation done on a large subset of the PDT. First row shows the number of sentences where a clause has alien tokens inside (precisely, where a clause is not bordered by conjunctions or punctuation). Second row shows the number of the problematic appositions whose governing token is a verb. Row three shows the number of sentences manifesting both phenomena. Evidently, the number of sentences where the intuitive and the definition-conforming splits of tokens to clauses differ is significant. However, the number of sentences where the algorithm fails to *do the right thing* (row 2) is almost negligible.

	Count	Ratio
Linearly incompact	7124	8.59%
Appositions	114	0.14%
incomp&appos	7225	8.71%
All	82944	100.00%

Table 1: Evaluation of the extraction of clauses from analytic trees.

Conclusions

Although the work described in this paper is bound to a particular language and to a particular treebank, we hope that the main achievement of our experiment is more general. It supports the claim that a consistently annotated treebank may provide even more information than primarily intended. Even complex linguistic phenomena may be extracted by means of relatively reliable methods.

The second most important result of our experiment are the actual data obtained as a result of application of our algorithm. They may serve for future experiments with complex Czech sentences and clauses. The lack of reliable data hindered the research of this very interesting phenomenon in the past. Our algorithm provides enough data not only for testing the theories, but also relatively enough training data allowing to apply not only the thorough linguistic investigation of the phenomenon, but also the application of modern stochastic methods.

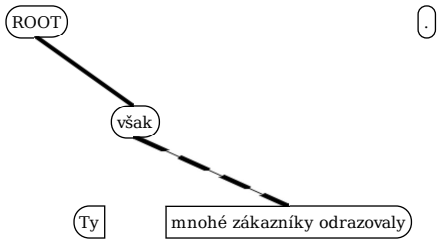

Figure 6: Sections of the sentence "Ty však mnohé zákazníky odrazovaly." [Those however many clients$_{obj}$ discouraged.] (Those have, however, discouraged many clients.) Here, the sentence is marked up as one coordinated clause, governed by the coordination formed by the "však" (however) conjunction. The reason for this maybe surprising annotation is that the sentence is de facto coordinated with the previous one.

Acknowledgments

This work was supported by the Grant Agency of the Czech Republic, grant No. 405/08/0681 and by the programme Information Society of the GAAV CR, grant No. 1ET100300517.

References

Hajič, J., Hladká, B.: *Probabilistic and Rule-Based Tagger of an Inflective Language – A Comparison* In: Proceedings of the Fifth Conference on Applied Natural Language Processing. Washington D.C., 111-118, 1997

Hajič, J., Hajičová, E., Pajas, P., Panevová, J., Sgall, P., Vidová-Hladká, B.: *Prague Dependency Treebank 1.0 (Final Production Label).* In: CD-ROM, Linguistic Data Consortium, 2001

Hajič, J., Panevová, J., Buráňová, E., Urešová, Z. Bémová, A.: *Anotace Pražského závislostního korpusu na analytické rovině: pokyny pro anotátory.* Technical Report No. 28, ÚFAL MFF UK, Prague, Czech Republic, 1999

Hajič, J.: *Building a Syntactically Annotated Corpus: The Prague Dependency Treebank* In Issues of Valency and Meaning, pp. 106-132, Karolinum, Praha 1998

Hajič, J., Panevová, J., Buráňová, E., Urešová, Z. and Bémová,A.: *A Manual for Analytic Layer Tagging of the Prague Dependency Treebank*, ISBN 1-58563-212-0, 2001

Hajičová, E. : *Prague Dependency Treebank: From Analytic to Tectogrammatical Annotations*. In: Text, Speech , Dialogue, ed. by P. Sojka, V. Matoušek and I. Kopeček, Brno, Masaryk University, 45-50, 1998

Hajičová, E. : *The Prague Dependency Treebank: Crossing the Sentence Boundary*. In: Text, Speech and Dialogue, ed. by V. Matoušek, P. Mautner, J. ocelíková and P. Sojka, Berlin: Springer, 20-27, 1999

Holan, T., Lopatková, M.: *Segmentation Charts for Czech – Relations among Segments in Complex Sentences*, submitted for LATA 2009

Kuboň, V., Lopatková, M., Plátek, M., Pognan, P.: *Segmentation of Complex Sentences*, In: Lecture Notes in Computer Science 4188, Text, Speech and Dialogue, TSD 2006, Springer Berlin / Heidelberg 2006, 151-158

Sgall, P., Hajičová, E., Panevová, J.: *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*, Springer, 1986 ISBN 9027718385, 9789027718389