PDT 2.0 Requirements on a Query Language

Jiří Mírovský
Institute of Formal and Applied Linguistics
Charles University in Prague
Malostranské nám. 25, 118 00 Prague 1, Czech Republic
mirovsky@ufal.mff.cuni.cz

Abstract

Linguistically annotated treebanks play an essential part in the modern computational linguistics. The more complex the treebanks become, the more sophisticated tools are required for using them, namely for searching in the data. We study linguistic phenomena annotated in the Prague Dependency Treebank 2.0 and create a list of requirements these phenomena set on a search tool, especially on its query language.

1 Introduction

Searching in a linguistically annotated treebank is a principal task in the modern computational linguistics. A search tool helps extract useful information from the treebank, in order to study the language, the annotation system or even to search for errors in the annotation.

The more complex the treebank is, the more sophisticated the search tool and its query language needs to be. The Prague Dependency Treebank 2.0 (Hajič et al. 2006) is one of the most advanced manually annotated treebanks. We study mainly the tectogrammatical layer of the Prague Dependency Treebank 2.0 (PDT 2.0), which is by far the most advanced and complex layer in the treebank, and show what requirements on a query language the annotated linguistic phenomena bring. We also add requirements set by lower layers of annotation.

In *section 1* (after this introduction) we mention related works on search languages for various types of corpora. Afterwards, we very shortly introduce PDT 2.0, just to give a general picture of the principles and complexion of the annotation scheme.

In *section 2* we study the annotation manual for the tectogrammatical layer of PDT 2.0 (t-manual, Mikulová et al. 2006) and collect linguistic phenomena that bring special requirements on the query language. We also study lower layers of annotation and add their requirements.

In *section 3* we summarize the requirements in an extensive list of features required from a search language.

We conclude in *section 4*.

1.1 Related Work

In Lai, Bird 2004, the authors name seven linguistic queries they consider important representatives for checking a sufficiency of a query language power. They study several query tools and their query languages and compare them on the basis of their abilities to express these seven queries. In Bird et al. 2005, the authors use a revised set of seven key linguistic queries as a basis for forming a list of three expressive features important for linguistic queries. The features are: immediate precedence, subtree scoping and edge alignment. In Bird et al. 2006, another set of seven linguistic queries is used to show a necessity to enhance XPath (a standard query language for XML, Clark, DeRose 1999) to support linguistic queries.

Cassidy 2002 studies adequacy of XQuery (a search language based on XPath, Boag et al. 1999) for searching in hierarchically annotated data. Re-

Proceedings of ACL-08: HLT, pages 37–45,
Columbus, Ohio, USA, June 2008.

quirements on a query language for annotation graphs used in speech recognition is also presented in Bird et al. 2000. A description of linguistic phenomena annotated in the Tiger Treebank, along with an introduction to a search tool TigerSearch, developed especially for this treebank, is given in Brants et al. 2002, nevertheless without a systematic study of the required features.

Laura Kallmeyer (Kallmeyer 2000) studies requirements on a query language based on two examples of complex linguistic phenomena taken from the NEGRA corpus and the Penn Treebank, respectively.

To handle alignment information, Merz and Volk 2005 study requirements on a search tool for parallel treebanks.

All the work mentioned above can be used as an ample source of inspiration, though it cannot be applied directly to PDT 2.0. A thorough study of the PDT 2.0 annotation is needed to form conclusions about requirements on a search tool for this dependency tree-based corpus, consisting of several layers of annotation and having an extremely complex annotation scheme, which we shortly describe in the next subsection.

1.2 The Prague Dependency Treebank 2.0

The Prague Dependency Treebank 2.0 is a manually annotated corpus of Czech. The texts are annotated on three layers – morphological, analytical and tectogrammatical.

On the morphological layer, each token of every sentence is annotated with a lemma (attribute `m/lemma`), keeping the base form of the token, and a tag (attribute `m/tag`), which keeps its morphological information.

The analytical layer roughly corresponds to the surface syntax of the sentence; the annotation is a single-rooted dependency tree with labeled nodes. Attribute `a/afun` describes the type of dependency between a dependent node and its governor. The order of the nodes from left to right corresponds exactly to the surface order of tokens in the sentence (attribute `a/ord`).

The tectogrammatical layer captures the linguistic meaning of the sentence in its context. Again, the annotation is a dependency tree with labeled nodes (Hajičová 1998). The correspondence of the nodes to the lower layers is often not 1:1 (Mírovský 2006).

Attribute `functor` describes the dependency between a dependent node and its governor. A tectogrammatical lemma (attribute `t_lemma`) is assigned to every node. 16 grammatemes (prefixed `gram`) keep additional annotation (e.g. `gram/verbmod` for verbal modality).

Topic and focus (Hajičová et al. 1998) are marked (attribute `tfa`), together with so-called deep word order reflected by the order of nodes in the annotation (attribute `deepord`).

Coreference relations between nodes of certain category types are captured. Each node has a unique identifier (attribute `id`). Attributes `coref_text.rf` and `coref_gram.rf` contain `id`s of coreferential nodes of the respective types.

2 Phenomena and Requirements

We make a list of linguistic phenomena that are annotated in PDT 2.0 and that determine the necessary features of a query language.

Our work is focused on two structured layers of PDT 2.0 – the analytical layer and the tectogrammatical layer. For using the morphological layer exclusively and directly, a very good search tool Manatee/Bonito (Rychlý 2000) can be used. We intend to access the morphological information only from the higher layers, not directly. Since there is relation 1:1 among nodes on the analytical layer (but for the technical root) and tokens on the morphological layer, the morphological information can be easily merged into the analytical layer – the nodes only get additional attributes.

The tectogrammatical layer is by far the most complex layer in PDT 2.0, therefore we start our analysis with a study of the annotation manual for the tectogrammatical layer (t-manual, Mikulová et al. 2006) and focus also on the requirements on accessing lower layers with non-1:1 relation. Afterwards, we add some requirements on a query language set by the annotation of the lower layers – the analytical layer and the morphological layer.

During the studies, we have to keep in mind that we do not only want to search for a phenomenon, but also need to study it, which can be a much more complex task. Therefore, it is not sufficient e.g. to find a predicative complement, which is a trivial task, since attribute `functor` of the complement is set to value `COMPL`. In this particular example, we also need to be able to specify in the

query properties of the node the second dependency of the complement goes to, e.g. that it is an Actor.

A summary of the required features on a query language is given in the subsequent section.

2.1 The Tectogrammatical Layer

First, we focus on linguistic phenomena annotated on the tectogrammatical layer. T-manual has more than one thousand pages. Most of the manual describes the annotation of simple phenomena that only require a single-node query or a very simple structured query. We mostly focus on those phenomena that bring a special requirement on the query language.

2.1.1 Basic Principles

The basic unit of annotation on the tectogrammatical layer of PDT 2.0 is a sentence.

The representation of the tectogrammatical annotation of a sentence is a rooted dependency tree. It consists of a set of nodes and a set of edges. One of the nodes is marked as a root. Each node is a complex unit consisting of a set of pairs attribute-value (t-manual, page 1). The edges express dependency relations between nodes. The edges do not have their own attributes; attributes that logically belong to edges (e.g. type of dependency) are represented as node-attributes (t-manual, page 2).

It implies the first and most basic requirement on the query language: one result of the search is one sentence along with the tree belonging to it. Also, the query language should be able to express node evaluation and tree dependency among nodes in the most direct way.

2.1.2 Valency

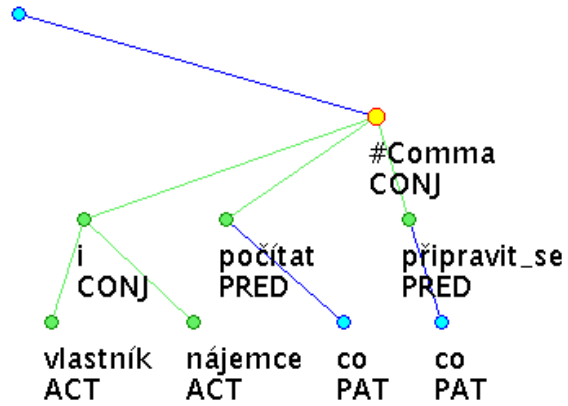
Valency of semantic verbs, valency of semantic verbal nouns, valency of semantic nouns that represent the nominal part of a complex predicate and valency of some semantic adverbs are annotated fully in the trees (t-manual, pages 162-3). Since the valency of verbs is the most complete in the annotation and since the requirements on searching for valency frames of nouns are the same as of verbs, we will (for the sake of simplicity in expressions) focus on the verbs only. Every verb meaning is assigned a valency frame. Verbs usually have more than one meaning; each is assigned a separate valency frame. Every verb has as many valency frames as it has meanings (t-manual, page 105).

Therefore, the query language has to be able to distinguish valency frames and search for each one of them, at least as long as the valency frames differ in their members and not only in their index. (Two or more identical valency frames may represent different verb meanings (t-manual, page 105).) The required features include a presence of a son, its non-presence, as well as controlling number of sons of a node.

2.1.3 Coordination and Apposition

Tree dependency is not always linguistic dependency (t-manual, page 9). Coordination and apposition are examples of such a phenomenon (t-manual, page 282). If a Predicate governs two coordinated Actors, these Actors technically depend on a coordinating node and this coordinating node depends on the Predicate. the query language should be able to skip such a coordinating node. In general, there should be a possibility to skip any type of node.

Skipping a given type of node helps but is not sufficient. The coordinated structure can be more complex, for example the Predicate itself can be coordinated too. Then, the Actors do not even belong to the subtree of any of the Predicates. In the following example, the two Predicates (PRED) are coordinated with conjunction (CONJ), as well as the two Actors (ACT). The linguistic dependencies go from each of the Actors to each of the Predicates but the tree dependencies are quite different:



In Czech: *S čím mohou vlastníci i nájemci počítat, na co by se měli připravit?*

In English: *What can owners and tenants expect, what they should get ready for?*

The query language should therefore be able to express the linguistic dependency directly. The information about the linguistic dependency is annotated in the treebank by the means of references, as well as many other phenomena (see below).

2.1.4 Idioms (Phrasemes) etc.

Idioms/phrasemes (idiomatic/phraseologic constructions) are combinations of two or more words with a fixed lexical content, which together constitute one lexical unit with a metaphorical meaning (which cannot be decomposed into meanings of its parts) (t-manual, page 308). Only expressions which are represented by at least two auto-semantic nodes in the tectogrammatical tree are captured as idioms (functor DPHR). One-node (one-auto-semantic-word) idioms are not represented as idioms in the tree. For example, in the combination "chlapec k pohledání" ("a boy to look for"), the prepositional phrase gets functor RSTR, and it is not indicated that it is an idiom.

Secondary prepositions are another example of a linguistic phenomenon that can be easily recognized in the surface form of the sentence but is difficult to find in the tectogrammatical tree.

Therefore, the query language should offer a basic searching in the linear form of the sentence, to allow searching for any idiom or phraseme, regardless of the way it is or is not captured in the tectogrammatical tree. It can even help in a situation when the user does not know how a certain linguistic phenomenon is annotated on the tectogrammatical layer.

2.1.5 Complex Predicates

A complex predicate is a multi-word predicate consisting of a semantically empty verb which expresses the grammatical meanings in a sentence, and a noun (frequently denoting an event or a state of affairs) which carries the main lexical meaning of the entire phrase (t-manual, page 345). Searching for a complex predicate is a simple task and does not bring new requirements on the query language. It is valency of complex predicates that requires our attention, especially dual function of a valency modification. The nominal and verbal components of the complex predicate are assigned the appropriate valency frame from the valency lexicon. By means of newly established nodes with `t_lemma` substitutes, those valency modification positions not present at surface layer are filled. There are problematic cases where the expressed valency modification occurs in the same form in the valency frames of both components of the complex predicate (t-manual, page 362).

To study these special cases of valency, the query language has to offer a possibility to define that a valency member of the verbal part of a complex predicate is at the same time a valency member of the nominal part of the complex predicate, possibly with a different function. The identity of valency members is annotated again by the means of references, which is explained later.

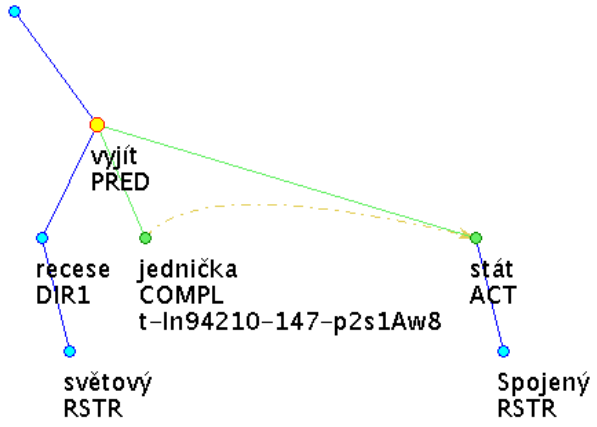
2.1.6 Predicative Complement (Dual Dependency)

On the tectogrammatical layer, also cases of the so-called predicative complement are represented. The predicative complement is a non-obligatory free modification (adjunct) which has a dual semantic dependency relation. It simultaneously modifies a noun and a verb (which can be nominalized).

These two dependency relations are represented by different means (t-manual, page 376):

- the dependency on a verb is represented by means of an edge (which means it is represented in the same way like other modifications),
- the dependency on a (semantic) noun is represented by means of attribute `compl.rf`, the value of which is the identifier of the modified noun.

In the following example, the predicative complement (COMPL) has one dependency on a verb (PRED) and another (dual) dependency on a noun (ACT):



In Czech: *Ze světové recese vyšly jako jednička Spojené státy.*
In English: *The United States emerged from the world recession as number one.*

The second form of dependency, represented once again with references (still see below), has to be expressible in the query language.

2.1.7 Coreferences

Two types of coreferences are annotated on the tectogrammatical layer:

- grammatical coreference
- textual coreference

The current way of representing coreference uses references (t-manual, page 996).

Let us finally explain what references are. References make use of the fact that every node of every tree has an identifier (the value of attribute `id`), which is unique within PDT 2.0. If coreference, dual dependency, or valency member identity is a link between two nodes (one node referring to another), it is enough to specify the identifier of the referred node in the appropriate attribute of the referring node. Reference types are distinguished by different referring attributes. Individual reference subtypes can be further distinguished by the value of another attribute.

The essential point in references (for the query language) is that at the time of forming a query, the value of the reference is unknown. For example, in the case of dual dependency of predicative complement, we know that the value of attribute `compl.rf` of the complement must be the same as the value of attribute `id` of the governing noun, but the value itself differs tree from tree and therefore is unknown at the time of creating the query. The query language has to offer a possibility to bind these unknown values.

2.1.8 Topic-Focus Articulation

On the tectogrammatical layer, also the topic-focus articulation (TFA) is annotated. TFA annotation comprises two phenomena:

- contextual boundness, which is represented by values of attribute `tfa` for each node of the tectogrammatical tree.
- communicative dynamism, which is represented by the underlying order of nodes.

Annotated trees therefore contain two types of information - on the one hand the value of contextual boundness of a node and its relative ordering with respect to its brother nodes reflects its function within the topic-focus articulation of the sentence, on the other hand the set of all the TFA values in the tree and the relative ordering of subtrees reflect the overall functional perspective of the sentence, and thus enable to distinguish in the sentence the complex categories of topic and focus (however, these are not annotated explicitly) (t-manual, page 1118).

While contextual boundness does not bring any new requirement on the query language, communicative dynamism requires that the relative order of nodes in the tree from left to right can be expressed. The order of nodes is controlled by attribute `deepord`, which contains a non-negative real (usually natural) number that sets the order of the nodes from left to right. Therefore, we will again need to refer to a value of an attribute of another node but this time with relation other than "equal to".

2.1.8.1 Focus Proper

Focus proper is the most dynamic and communicatively significant contextually non-bound part of the sentence. Focus proper is placed on the rightmost path leading from the effective root of the tectogrammatical tree, even though it is at a different position in the surface structure. The node representing this expression will be placed rightmost in the tectogrammatical tree. If the focus proper is constituted by an expression represented as the effective root of the tectogrammatical tree (i.e. the governing predicate is the focus proper), there is no right path leading from the effective root (t-manual, page 1129).

2.1.8.2 Quasi-Focus

Quasi-focus is constituted by (both contrastive and non-contrastive) contextually bound expressions, on which the focus proper is dependent. The focus proper can immediately depend on the quasi-focus, or it can be a more deeply embedded expression.

In the underlying word order, nodes representing the quasi-focus, although they are contextually bound, are placed to the right from their governing node. Nodes representing the quasi-focus are therefore contextually bound nodes on the rightmost

path in the tectogrammatical tree (t-manual, page 1130).

The ability of the query language to distinguish the rightmost node in the tree and the rightmost path leading from a node is therefore necessary.

2.1.8.3 Rhematizers

Rhematizers are expressions whose function is to signal the topic-focus articulation categories in the sentence, namely the communicatively most important categories - the focus and contrastive topic.

The position of rhematizers in the surface word order is quite loose, however they almost always stand right before the expressions they rhematize, i.e. the expressions whose being in the focus or contrastive topic they signal (t-manual, pages 1165-6).

The guidelines for positioning rhematizers in tectogrammatical trees are simple (t-manual, page 1171):

- a rhematizer (i.e. the node representing the rhematizer) is placed as the closest left brother (in the underlying word order) of the first node of the expression that is in its scope.
- if the scope of a rhematizer includes the governing predicate, the rhematizer is placed as the closest left son of the node representing the governing predicate.
- if a rhematizer constitutes the focus proper, it is placed according to the guidelines for the position of the focus proper - i.e. on the rightmost path leading from the effective root of the tectogrammatical tree.

Rhematizers therefore bring a further requirement on the query language – an ability to control the distance between nodes (in the terms of deep word order); at the very least, the query language has to distinguish an immediate brother and relative horizontal position of nodes.

2.1.8.4 (Non-)Projectivity

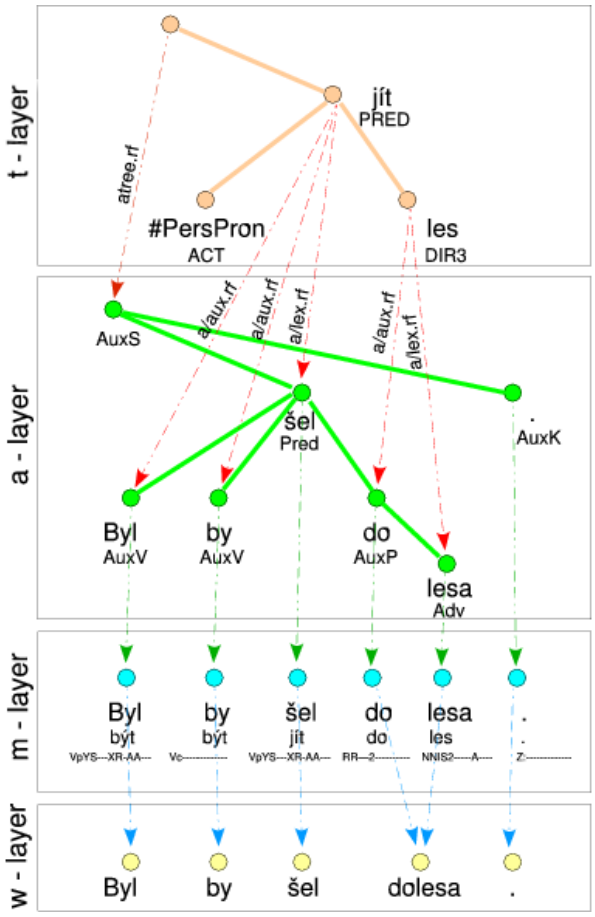
Projectivity of a tree is defined as follows: if two nodes B and C are connected by an edge and C is to the left from B, then all nodes to the right from B and to the left from C are connected with the root via a path that passes through at least one of the nodes B or C. In short: between a father and its son there can only be direct or indirect sons of the father (t-manual, page 1135).

The relative position of a node (node A) and an edge (nodes B, C) that together cause a non-projectivity forms four different configurations: (“B is on the left from C” or “B is on the right from C”) x (“A is on the path from B to the root” or “it is not”). Each of the configurations can be searched for using properties of the language that have been required so far by other linguistic phenomena. Four different queries search for four different configurations.

To be able to search for all configurations in one query, the query language should be able to combine several queries into one multi-query. We do not require that a general logical expression can be set above the single queries. We only require a general OR combination of the single queries.

2.1.9 Accessing Lower Layers

Studies of many linguistic phenomena require a multilayer access.



In Czech: *Byl by šel do lesa.*

In English (lit.): *He would have gone to the forest.*

For example, the query "find an example of Patient that is more dynamic than its governing Predicate (with greater `deepord`) but on the surface layer is on the left side from the Predicate" requires information both from the tectogrammatical layer and the analytical layer.

The picture above is taken from PDT 2.0 guide and shows the typical relation among layers of annotation for the sentence (the lowest w-layer is a technical layer containing only the tokenized original data).

The information from the lower layers can be easily compressed into the analytical layer, since there is relation 1:1 among the layers (with some rare exceptions like misprints in the w-layer). The situation between the tectogrammatical layer and the analytical layer is much more complex. Several nodes from the analytical layer may be (and often are) represented by one node on the tectogrammatical layer and new nodes without an analytical counterpart may appear on the tectogrammatical layer. It is necessary that the query language addresses this issue and allows access to the information from the lower layers.

2.2 The Analytical and Morphological Layer

The analytical layer is much less complex than the tectogrammatical layer. The basic principles are the same – the representation of the structure of a sentence is rendered in the form of a tree – a connected acyclic directed graph in which no more than one edge leads into a node, and whose nodes are labeled with complex symbols (sets of attributes). The edges are not labeled (in the technical sense). The information logically belonging to an edge is represented in attributes of the depending node. One node is marked as a root.

Here, we focus on linguistic phenomena annotated on the analytical and morphological layer that bring a new requirement on the query language (that has not been set in the studies of the tectogrammatical layer).

2.2.1 Morphological Tags

In PDT 2.0, morphological tags are positional. They consist of 15 characters, each representing a certain morphological category, e.g. the first position represents part of speech, the third position represents gender, the fourth position represents number, the fifth position represents case.

The query language has to offer a possibility to specify a part of the tag and leave the rest unspecified. It has to be able to set such conditions on the tag like "this is a noun", or "this is a plural in fourth case". Some conditions might include negation or enumeration, like "this is an adjective that is not in fourth case", or "this is a noun either in third or fourth case". This is best done with some sort of wild cards. The latter two examples suggest that such a strong tool like regular expressions may be needed.

2.2.2 Agreement

There are several cases of agreement in Czech language, like agreement in case, number and gender in attributive adjective phrase, agreement in gender and number between predicate and subject (though it may be complex), or agreement in case in apposition.

To study agreement, the query language has to allow to make a reference to only a part of value of attribute of another node, e.g. to the fifth position of the morphological tag for case.

2.2.3 Word Order

Word order is a linguistic phenomenon widely studied on the analytical layer, because it offers a perfect combination of a word order (the same like in the sentence) and syntactic relations between the words. The same technique like with the deep word order on the tectogrammatical layer can be used here. The order of words (tokens) ~ nodes in the analytical tree is controlled by attribute `ord`. Non-projective constructions are much more often and interesting here than on the tectogrammatical layer. Nevertheless, they appear also on the tectogrammatical layer and their contribution to the requirements on the query language has already been mentioned.

The only new requirement on the query language is an ability to measure the horizontal distance between words, to satisfy linguistic queries like "find trees where a preposition and the head of the noun phrase are at least five words apart".

3 Summary of the Features

Here we summarize what features the query language has to have to suit PDT 2.0. We list the features from the previous section and also add some

obvious requirements that have not been mentioned so far but are very useful generally, regardless of a corpus.

3.1 Complex Evaluation of a Node

- multiple attributes evaluation (an ability to set values of several attributes at one node)
- alternative values (e.g. to define that `functor` of a node is either a disjunction or a conjunction)
- alternative nodes (alternative evaluation of the whole set of attributes of a node)
- wild cards (regular expressions) in values of attributes (e.g. `m/tag="N...4.*"` defines that the morphological tag of a node is a noun in accusative, regardless of other morphological categories)
- negation (e.g. to express "this node is not Actor")
- relations less than (<=) , greater than (>=) (for numerical attributes)

3.2 Dependencies Between Nodes (Vertical Relations)

- immediate, transitive dependency (existence, non-existence)
- vertical distance (from root, from one another)
- number of sons (zero for lists)

3.3 Horizontal Relations

- precedence, immediate precedence, horizontal distance (all both positive, negative)
- secondary edges, secondary dependencies, coreferences, long-range relations

3.4 Other Features

- multiple-tree queries (combined with general OR relation)
- skipping a node of a given type (for skipping simple types of coordination, apposition etc.)
- skipping multiple nodes of a given type (e.g. for recognizing the rightmost path)
- references (for matching values of attributes unknown at the time of creating the query)
- accessing several layers of annotation at the same time with non-1:1 relation (for studying relation between layers)
- searching in the surface form of the sentence

4 Conclusion

We have studied the Prague Dependency Treebank 2.0 tectogrammatical annotation manual and listed linguistic phenomena that require a special feature from any query tool for this corpus. We have also added several other requirements from the lower layers of annotation. We have summarized these features, along with general corpus-independent features, in a concise list.

Acknowledgment

This research was supported by the Grant Agency of the Academy of Sciences of the Czech Republic, project IS-REST (No. 1ET101120413).

References

Bird et al. 2000. Towards A Query Language for Annotation Graphs. *In: Proceedings of the Second International Language and Evaluation Conference, Paris, ELRA, 2000.*

Bird et al. 2005. Extending Xpath to Support Linguistc Queries. *In: Proceedings of the Workshop on Programming Language Technologies for XML, California, USA, 2005.* .

Bird et al. 2006. Designing and Evaluating an XPath Dialect for Linguistic Queries. *In: Proceedings of the 22nd International Conference on Data Engineering (ICDE), pp 52-61, Atlanta, USA, 2006.*

Boag et al. 1999. XQuery 1.0: An XML Query Language. *IW3C Working Draft, http://www.w3.org/TR/xpath, 1999.*

Brants S. et al. 2002. The TIGER Treebank. *In: Proceedings of TLT 2002, Sozopol, Bulgaria, 2002.*

Cassidy S. 2002. XQuery as an Annotation Query Language: a Use Case Analysis. *In: Proceedings of the Third International Conference on Language Resources and Evaluation, Canary Islands, Spain, 2002*

Clark J., DeRose S. 1999. XML Path Language (XPath). *http://www.w3.org/TR/xpath, 1999.*

Hajič J. et al. 2006. Prague Dependency Treebank 2.0. *CD-ROM LDC2006T01, LDC, Philadelphia, 2006.*

Hajičová E. 1998. Prague Dependency Treebank: From analytic to tectogrammatical annotations. *In: Proceedings of 2nd TST, Brno, Springer-Verlag Berlin Heidelberg New York, 1998, pp. 45-50.*

Hajičová E., Partee B., Sgall P. 1998. Topic-Focus Articulation, Tripartite Structures and Semantic Content. *Dordrecht, Amsterdam, Kluwer Academic Publishers, 1998.*

Havelka J. 2007. Beyond Projectivity: Multilingual Evaluation of Constraints and Measures on Non-Projective Structures. *In Proceedings of ACL 2007, Prague, pp. 608-615.*

Kallmeyer L. 2000: On the Complexity of Queries for Structurally Annotated Linguistic Data. *In Proceedings of ACIDCA'2000, Corpora and Natural Language Processing, Tunisia, 2000, pp. 105-110.*

Lai C., Bird S. 2004. Querying and updating treebanks: A critical survey and requirements analysis. *In: Proceedings of the Australasian Language Technology Workshop, Sydney, Australia, 2004*

Merz Ch., Volk M. 2005. Requirements for a Parallel Treebank Search Tool. *In: Proceedings of GLDV-Conference, Bonn, Germany, 2005.*

Mikulová et al. 2006. Annotation on the Tectogrammatical Level in the Prague Dependency Treebank (Reference Book). *ÚFAL/CKL Technical Report TR-2006-32, Charles University in Prague, 2006.*

Mírovský J. 2006. Netgraph: a Tool for Searching in Prague Dependency Treebank 2.0. *In Proceedings of TLT 2006, Prague, pp. 211-222.*

Rychlý P. 2000. Korpusové manažery a jejich efektivní implementace. *PhD. Thesis, Brno, 2000.*