Intelligent Information Systems 9999
ISBN 666-666-666, pages 1–12

Dendrarium—an Open Source Tool for Treebank Building

Marcin Woliński

Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

Abstract

This paper presents a new web-based tool for treebank development. The tool provides an environment for disambiguation and validation of parse forests generated by a DCG grammar. Disambiguation proceeds in a top-down manner. Every sentence is processed independently by two users. Feedback from the system is used to improve the grammar, which leads to iterative parallel grammar and treebank development. An important feature of the system is minimisation of repeated work caused by the changes in the grammar.

Keywords: treebank of Polish, constituent parse trees, grammar development

1 Introduction

The tool being presented here has been developed within a project whose aim is to build a treebank of Polish.[1] To the best of our knowledge it will be the first large treebank of Polish. In the first phase it will contain trees for about 20,000 sentences (cf. Świdziński and Woliński, to appear).

The treebank is being built in a semi-automatic process. Parse trees are generated by a parser and then selected and validated by human annotators. The process is iterative, since the grammar undergoes improvements based on the feedback from the annotators.

We work on one million word balanced sub-corpus of the National Corpus of Polish (NKJP, `http://nkjp.pl`, Przepiórkowski *et al.* (2010, 2008)). This sub-corpus has been manually annotated with morphological features, which means that the results of a morphological analyser were disambiguated and descriptions for words unknown to the analyser (mainly proper names) were added. Consequently every word in this corpus has one morphological interpretation.

The grammar used in the project is a new version of Marek Świdziński's DCG grammar of Polish (Świdziński, 1992) implemented by Woliński (2004). Since then the grammar has undergone a deep reconstruction. The set of nonterminals has been limited (e.g., seven clause types have been reduced to one), resulting in much simpler and 'flatter' trees. We are in the process of adding to the grammar the long missed constructs, such as coordinated phrases of various types (Świdziński

[1] The project is partially funded by the research grant N N104 224735 from the Polish Ministry of Science and Higher Education.

and Woliński, 2009). Another interesting new feature is that in every rule one of the constituent phrases is marked as a syntactic centre, thus although we work with constituent trees, it is possible to generate dependence trees from them as well (cf. Nivre, 2003). On the downside, working with a constituent grammar means we cannot draw much experience from other projects for Slavic languages using dependency formalisms (most notably the PDT (Böhmová *et al.*, 2003)).

As the aim of the Świdziński's grammar is to catch all possible structures, and the parser does not include any statistical component, it is typical to get many trees even for simple sentences. The parse forests are disambiguated manually.

2 Dendrarium

To facilitate the task of disambiguation and validation of parse forests we have developed a system named Dendrarium. The work is being done by a group of users, so it seemed natural to use a web based application.

Technically speaking the system has been implemented in the Jboss Seam framework, so a Java application is running on the server with PostgreSQL as a database engine. On the client machine a web browser displays HTML forms with some JavaScript elements. The system has been targeted at the Firefox browser.

From the start Dendrarium is being developed as an open source application with the complete source code available at `http://sourceforge.net/projects/dendrarium`.[2]

The trees are processed in an iterative process: a linguist tries to determine the right tree for each sentence. If a valid tree is not present in the forest, the linguist can judge the sentence ungrammatical in which case the processing of the given sentence ends. But if the sentence is judged correct, it is examined by the authors of the grammar, who will improve the grammar and the sentence will be analysed once again.

Thus the system has to maintain the forests, the selected trees, and the assessments of the grammaticality of sentences in order to allow replacing parse forests with new versions without the need to repeat the whole work from the start.

We have identified the following roles of users of the system:

dendrologist – selects and validates parse trees,

superdendrologist – judges conflicts between dendrologists,

grammarian – inspects sentences pointed by dendrologists to improve their analysis by the parser,

administrator – manages users, loads (new versions of) parse forests to the system.

The role of an administrator although important is least interesting here. The other roles will be discussed in detail in the next section.

[2]The implementors of Dendrarium are students at Warsaw University Karolina Sołtys, Piotr Achinger, Andrzej Zaborowski, Dominika Pawlik and Tomasz Badowski.

3 The Processing of a Sentence in Dendrarium

Following the common practise, each sentence in Dendrarium is processed by two annotators (dendrologists) independently. If they give the same answer, it is considered valid (although the superdendrologist can inspect and change it). If a difference is detected by the system, each of the two dendrologists is asked to check his or her answer again, still not knowing the other answer. After this procedure, if the difference is still present, the sentence is marked as a conflict and passed to a superdendrologist, who will accept one of the answers given, or give a new one.

When seeking for the right tree for a given sentence the dendrologist has to pay attention to the following factors:

- Is each clause and phrase correctly divided into subclauses or subphrases?
- Has the finite verb in each clause been assigned the right arguments?
- Is the right element of each clause and phrase marked as the centre?
- Is the right morphological interpretation selected for each token of the sentence?

If the dendrologist decides that the correct tree is not present in the forest, they should select one of the so called 'special answers':

1. the utterance is incorrect (typos, punctuation errors, grammatical errors, etc.),
2. an error is present in morphological description or in segmentation,
3. the utterance is not a sentence,
4. sentence is correct, but too difficult to describe,
5. sentence is correct, grammar improvement requested.

This classification will allow us later to delve into selected problems. Due to limited resources in the current project we have decided not to correct in any way the texts or their morphological interpretations we receive from the NKJP project. Parsing such problematic utterances is a possible follow-up work. Similarly, we have decided to describe only sentences built around finite forms of verbs, leaving other utterances for later.

We also don't assume that we will be able to process every correct sentence. For example, since we use a constituent-based formalism non-contiguous phrases pose a problem. In the present project we describe only a very limited set of types of such phrases, which unfortunately means that most of such sentences will not get parse trees.

The special answer 'no correct tree' will cause the sentence to be looked at by the authors of the grammar.

The answers 1–4 are final in the sense that the sentence will not be processed any further and will not be included in the treebank. The answer 5 causes the sentence to be reprocessed when a new version of the grammar is created.

This leads to an iterative process advertised e.g. by Branco (2009). We want, however, to minimise the number of times a dendrologist is required to examine the same sentence. For that reason we present sentences to dendrologists only when we have some parse trees for them. If the dendrologist decides the set does

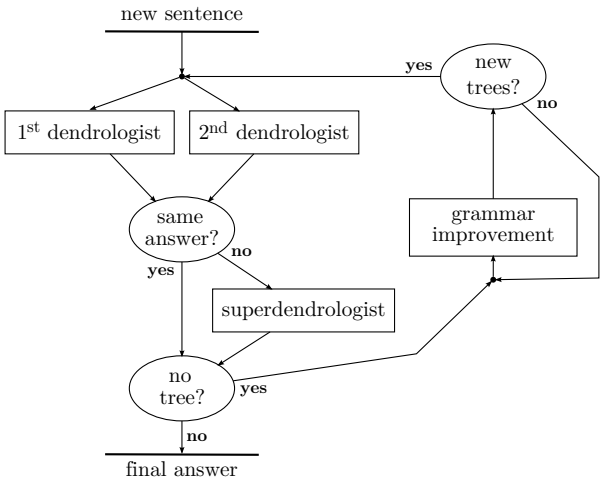

FIGURE 1: Processing of a sentence in Dendrarium

not include the proper tree, we will improve the grammar and present them with a new set of trees for that sentence (cf. Fig. 1). However, if the dendrologist accepts a tree, we will not update the forest for the given sentence even when a new version of the grammar starts to produce a different set of trees for that sentence. Only at the very end of feeding the system with sentences we will replace all forests with the version generated by the final grammar, and only then ask dendrologists to once more check the trees they selected.

But even then the dendrologists will not have to start their work anew (cf. Simov, 2003). The system includes a module which is responsible for 'smart' updating of forests. In particular, if a tree exactly identical to the manually selected tree is present in the new forest, we assume the tree remains the right answer even if the new forest includes more trees. If the tree is not there, the dendrologist has to reconsider the sentence, but even then the system will identify subtrees in the new forest identical to the selected subtrees in the old one and suggest them to the dendrologist.

If the dendrologist did the work well, and the tree selected was indeed the right structure for the sentence in question, the selected tree should not generally disappear in newer versions of the grammar. We plan to catch such cases automatically, as indicating possible new problems in the grammar, and pass them to grammarians for examination.

As can be seen, our approach differs from that of Rosén *et al.* (2006), where the identification of a tree is based on discriminants (questions about features not common to all trees in a forest). When grammar is enriched, the set of discriminants which used to select one tree may start to admit more trees (including the originally selected one). So after a grammar change the set of discriminants may need adjustments too. We avoid unnecessary work in that case. One weakness of this approach is that we are more dependent on the quality of the answer given by a dendrologist (is the selected tree really the right one?).

4 Selecting the Right Tree

The most important task of dendrologists is to determine the 'right' tree for each sentence. Parse forests are often large even for short sentences. They can easily include thousands of trees even after morphological disambiguation. We should however note, that this is a combinatorial explosion of only a few places in the structure where a subtree can be realized in several ways.

Obviously to make selection effective we cannot show every tree to a dendrologist and ask 'is this the correct one?'.[3] One possibility is to use discriminants. The easiest questions of this type are about morphological features of the words. But these have been already answered in the current project, since we have the disambiguated input. Asking meaningful questions about structural features is harder, so we have taken a different approach.

We concentrate on what we call ambiguous nodes in the parse forest. The only source of ambiguity in a DCG grammar is a situation when some nonterminal (with some fixed values of its attributes and spanning a fixed fragment of text) can be realised in more than one way by grammar rules. In such situation a subtree rooted at this node can be freely replaced with some other subtrees with the same nonterminal as root.

Our idea is to find an ambiguous node which is the closest to the root, and ask the dendrologist which of its possible realisations is the right one. The differences are generally meaningful from the linguistic point of view—a different set of constituents, a different partition of the respective fragment of the sentence—so we can reasonably expect the dendrologist to be able to answer such a question. When the node is disambiguated, we search for another ambiguous node and so on. The process resembles building the tree top-down, but the unambiguous fragments are filled in automatically and the dendrologist only selects from available options.

Since we have morphological disambiguation available, the ambiguous nodes are limited to purely structural problems. The most typical of them is the set and the extents of the dependants of a verb. Another is the attachment of modifiers within phrases. In coordinated clauses there may be various possible ways of grouping. And so on.

Figure 2 shows an example question asked by the system. In this case a nominal phrase consists of two nouns in the genitive and an adjective, so each of the nouns can be a modifier for the other one. In the two possible variants the system highlights the word which is the centre of the phrase (*przemiany* 'transformation' or *cudu* 'miracle') and the value of gender in the attributes of the phrase (in Polish 'transformation' is feminine and 'miracle' is masculine inanimate). In this case it is the miracle that was performed, so we should select the second variant. Answering this single question is enough to disambiguate the sentence (cf. Fig. 4).

For each variant the complete set of attributes is shown for each nonterminal of the right hand side of the corresponding grammatical rule. (Here only one, since a 'required phrase' fw is realised by a 'nominal phrase' fno.) If the information

[3] This one reason is unfortunately enough to dismiss the otherwise very interesting tool Tred (Hajič *et al.*, 2001).

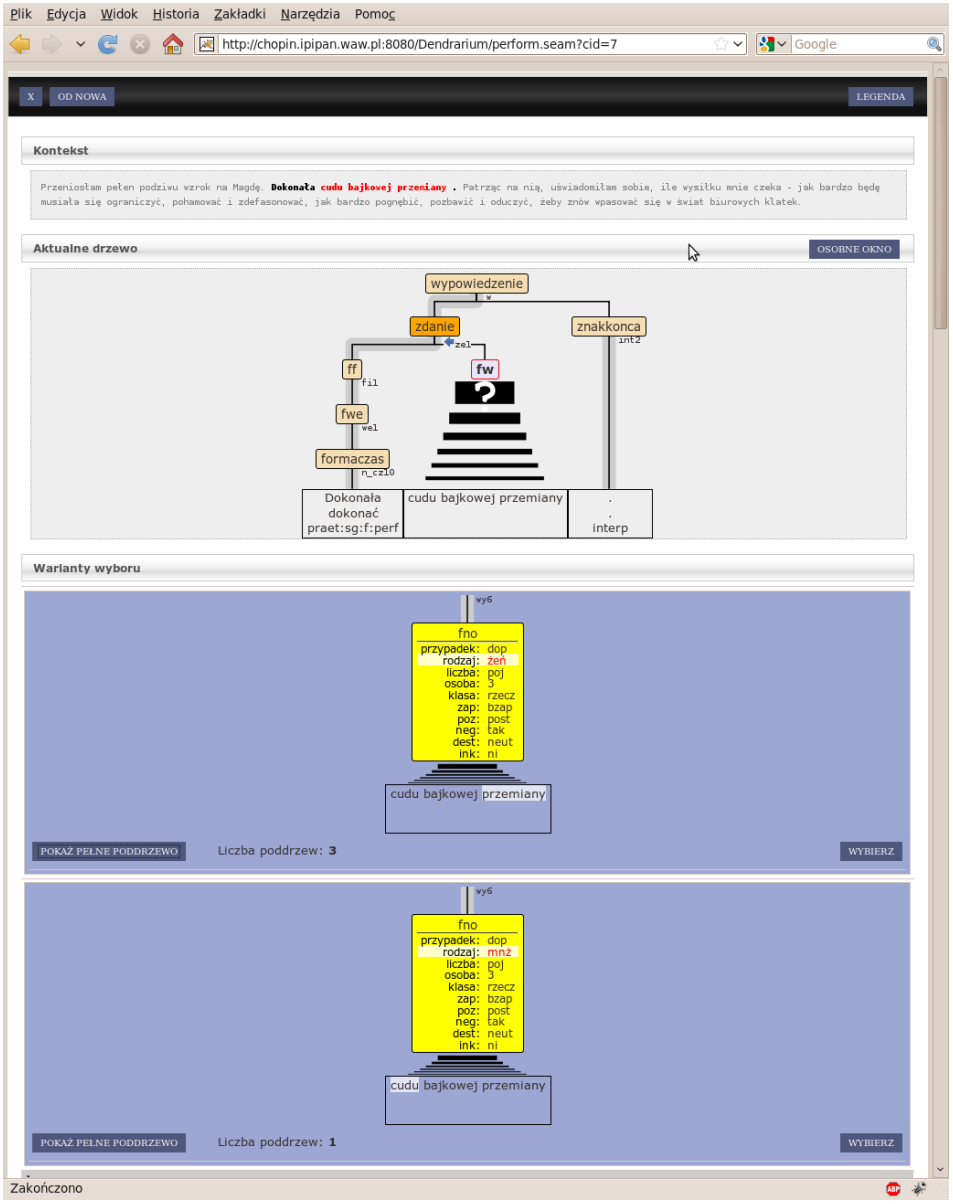

FIGURE 2: A partially disambiguated tree for the sentence *Dokonała cudu bajkowej przemiany* (‘*[She] made a miracle of a fairytale-like transformation.*’). Below the tree two possible variants for the ambiguous nominal phrase are shown.

provided for a variant is not sufficient to make a decision, the dendrologist can press the button 'show complete subtrees' for any particular variant and browse through all subtrees corresponding to the given realisation of the phrase in question (see Fig. 3).

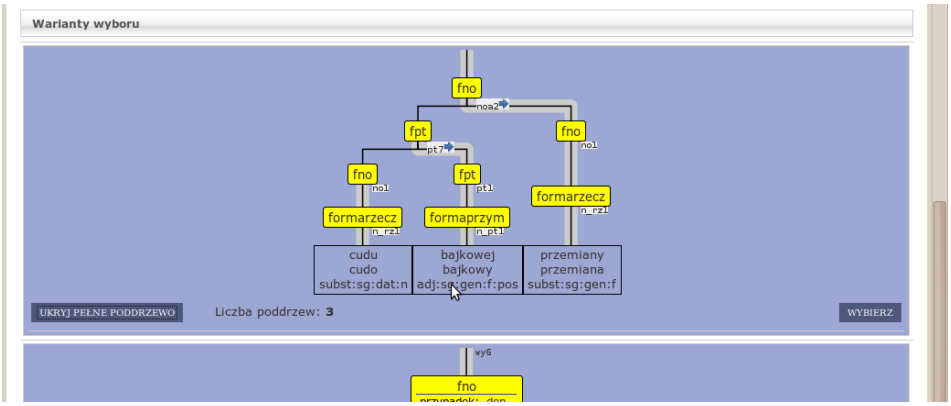

FIGURE 3: One of subtrees for the nominal phrase *cudu bajkowej przemiany*, where 'transformation' is the centre of the phrase.

We think this process is more intuitive than the bottom-up procedure used in the Annotate tool (Brants and Plaehn, 2000). It seems that when building the tree bottom-up the user anyway has to build the upper parts of the tree in his mind, to make sure that the part being built really fits into some larger structure. In our approach it is the system that maintains this feature: if the tree fragment shown is correct, there exists a complete tree containing that fragment.

Incremental disambiguation of ambiguous nodes is similar to disambiguation with tree intersections as used in the CLaRK system for HPSG grammars (cf. Simov *et al.*, 2002). However, working with a simpler formalism, we are able to compute the ambiguous nodes directly without ever generating all trees. Our parser works internally with shared/packed parse forests (Billot and Lang, 1989)—a structure similar to charts used in chart-parsing. In these structures tree nodes are represented directly, ambiguous ones having more than one possible set of children. The structures are polynomial in size even for a grammar which generates exponential number of interpretations. On the practical size this often corresponds to a number of computed structures several orders of magnitude lower than the number of trees.

The manual disambiguation of morphological forms provided by NKJP is automatically taken into account by the system. However, since we assume the annotators of NKJP can be wrong, the parser works on all possible interpretations and only then its results are filtered within Dendrarium. The user is presented only with the trees which use the disambiguated interpretations. However, if the correct tree is not among them, the dendrologist can switch to the 'all trees' mode.

Even though the Świdziński's grammar describes only syntax, the dendrologist has to take meaning into account when making their decisions. For example in the

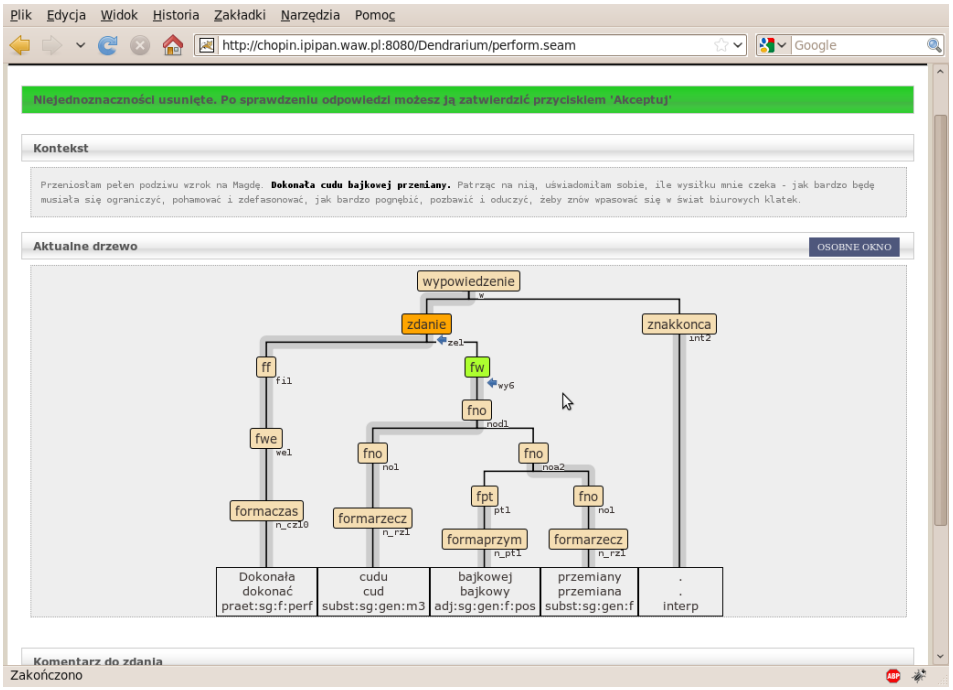

FIGURE 4: A complete disambiguated parse tree

sentence *Będzie projektował gmach Muzeum Historii Żydów Polskich w Warszawie.* ('*[He] will design a building for the Museum of the History of Polish Jews in Warsaw.*') the dendrologist has to decide, where to attach the modifier *in Warsaw*. The possibilities are that the unnamed designer will work in Warsaw, or the building will be in Warsaw, or the Museum is in Warsaw, or it is Polish Jews in Warsaw whose life is documented. In this case we decide to attach *in Warsaw* to *Museum*.

When the forest is disambiguated, i.e. a single tree is selected, it is the dendrologist's duty to double check whether the structure is correct, especially in parts which were filled-in automatically without asking questions. At this stage the user can still click any node in the tree to check available variants for that node and, possibly, change some selections.

All sentences in a given paragraph are assigned to the same dendrologist. Dendrologists are asked to take into account all knowledge they can get from this context, even if it is not present in a particular sentence. This is in accordance with the way morphological description was done in NKJP, where e.g., the gender of some pronouns was disambiguated on the basis of other sentences in the paragraph.

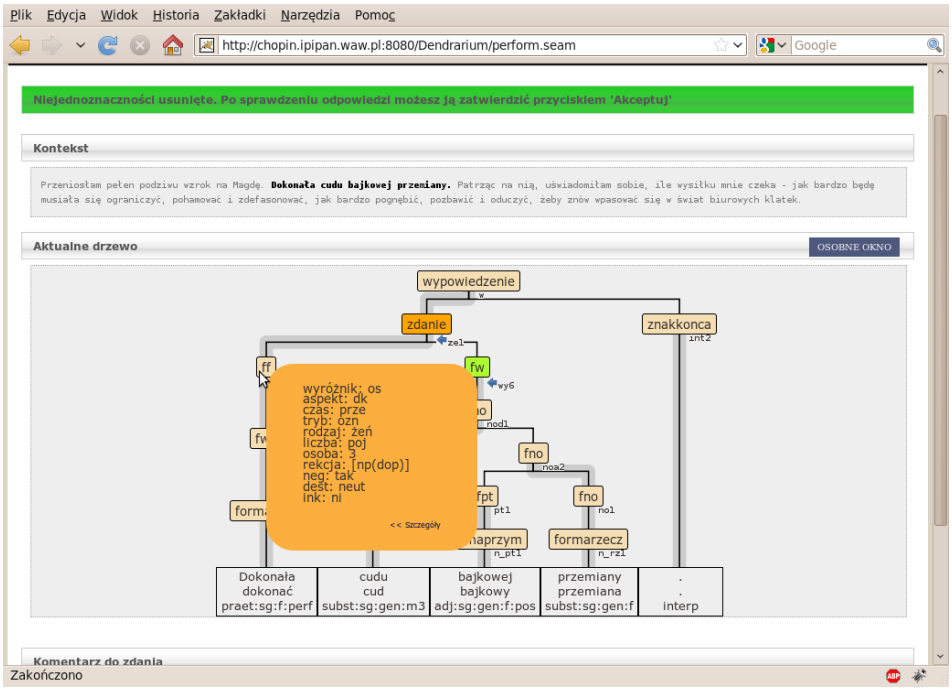

FIGURE 5: Attributes of the node ff displayed in an overlay

5 Visualisation of Parse Trees/Forests

An example of a complete tree rendered by Dendrarium is shown in figure 4. As is customary, the tree is displayed growing down, with the root on top and the tokens of the sentence on bottom.

As mentioned before, the text being analysed comes from the National Corpus of Polish, so the tokens and their morphological description correspond directly to that corpus. Each box representing a token contains: the word (preserving case), the lemma, and the morphological interpretation in the IPI PAN Tagset notation (cf. Przepiórkowski, 2009; Przepiórkowski and Woliński, 2003).

The nodes of the tree contain names of nonterminal units. The first layer above terminals contains units like formarzecz (nominal form), formaczas (verbal form), and so on. This level is used to represent the so called syntactic words in the corpus (see Przepiórkowski, 2008). Typically they correspond to just one token. However, we use them also to cater for analytical forms of verbs and other parts of speech (e.g., *będzie robić* 'will do') and some idiomatic expressions (e.g., *na pewno* 'surely').

The second layer of nonterminal units represents constituent phrases of various sorts: verbal phrases fwe, nominal phrases fno, adjectival phrases fpt, and so on. Obviously these can have arbitrary level of complication.

The third layer classifies phrases as constituents of clauses. Here we have a finite phrase ff, which will become the centre of a clause, and its dependants:

required phrases (arguments) fw and free phrases (adjuncts) fl.

The fourth layer comprises clauses. In the example we have only a simple clause with one verb and arguments, but clauses can get more complicated when coordination comes into play.

Finally, the root of the tree is utterance (wypowiedzenie), which consists of a clause and a final punctuation (znakkonca). As can be seen, punctuation characters are treated as constituents in the tree.

The Świdziński's grammar assigns numerous attributes to nonterminals. These attributes are not shown in the tree to conserve space. However, when a node is pointed to with the mouse, its attributes are displayed in a balloon overlay (cf. Fig. 5).

The thick gray shadows emphasising some branches in the tree show the centres of phrases. They join each of the nonterminals with the token which is its centre. This way the utterance and the clause in the example are connected with the verb *dokonać*, being the centre of the whole sentence. The node fno for nominal phrase *cudu bajkowej przemiany* is connected with the token *cudu*.

Trees are visualised using a cross-browser JavaScript library developed by Andrzej Zaborowski as part of his master's thesis (under preparation). For the sake of portability the library renders textual elements as absolutely positioned HTML frames, and for graphical elements resorts to SVG or VML depending on the browser. Thanks to this it should be possible to adapt Dendrarium to Internet Explorer if such a need arises.

The JavaScript library is able to visualise a single tree, but also a forest as a whole. In that mode one tree is shown at a time, but each ambiguous node is equipped with a couple of arrows (located near the rule symbol for the node) which allow to switch the view to the next/previous variant of realisation of the given node. This mechanism is a JavaScript realisation of the idea presented in Woliński 2006. Figure 6 presents an alternative structure of the example sentence based on a different interpretation of the word *cudu* (rejected by the annotators of NKJP).

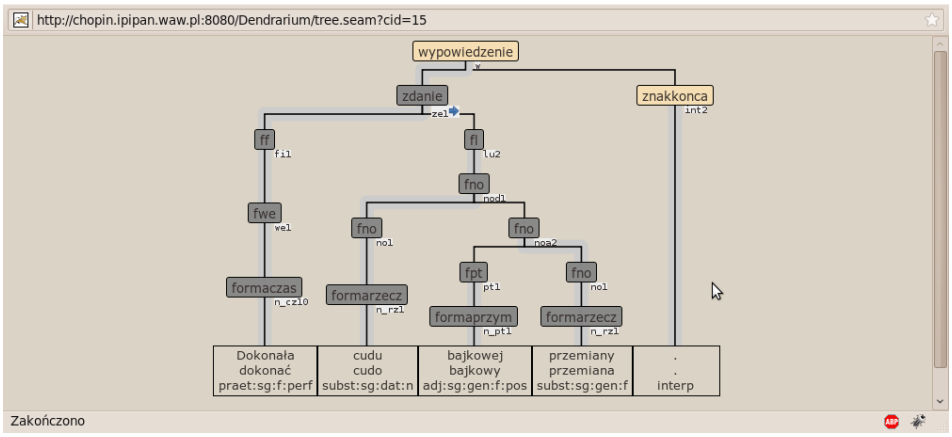

FIGURE 6: One of alternative trees for the same example sentence.

6 Conclusions

Since building of an actual treebank for Polish has only just started, we cannot yet provide data on the effectiveness of the process, the inter-annotator agreement, etc. The initial experiments have shown that the list of variants for the node zdanie (clause) is quite often difficult to disambiguate because of the large number of variants. Disambiguating this node in fact means selecting the right valence frame for the verb. We intend to add a filtering step to limit the number of variants shown at this point.

Dendrarium has been developed for a particular project, however it is not bound to a particular grammar or parser. The system makes rather minimal assumptions of the annotation used: constituent trees with nodes decorated with key/value pairs. The parse trees need to be represented as shared in an ad-hoc XML format, which should be easy to generate from other parsers. The availability of morphological disambiguation is desirable, but not really required. With the possibility to selectively ignore disambiguation even not completely reliable disambiguation (e.g., coming from a statistical tagger) can be used. An important feature of the system is that interaction happens in a standard web browser, so we do not require users, who are of linguistic background, to install non-standard software or to work in a particular platform. (In that respect Dendrarium is similar to the TREPIL tool (Rosén *et al.*, 2006) and differs from Annotate (Brants and Plaehn, 2000).)

References

Sylvie Billot and Bernard Lang (1989), The Structure of Shared Forests in Ambiguous Parsing, in *Meeting of the Association for Computational Linguistics*, pp. 143–151.

Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká (2003), The Prague Dependency Treebank: A 3-Level Annotation Scenario, in Anne Abeillé, editor, *Treebanks. Building and Using Parsed Corpora*, chapter 7, pp. 103–127, Kluwer Academic Publishers.

António Branco (2009), LogicalFormBanks, the Next Generation of Semantically Annotated Corpora: Key Issues in Construction Methodology, in Mieczysław A. Kłopotek, Adam Przepiórkowski, Sławomir T. Wierzchoń, and Krzysztof Trojanowski, editors, *Recent Advances in Intelligent Information Systems*, pp. 3–11, Akademicka Oficyna Wydawnicza Exit, Warsaw.

Thorsten Brants and Oliver Plaehn (2000), Interactive Corpus Annotation, in *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC 2000)*, Athens, Greece, URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.68.4540&rep=rep1&type=pdf`.

Jan Hajič, Barbora Vidová-Hladká, and Petr Pajas (2001), The Prague Dependency Treebank: Annotation Structure and Support, in *Proceedings of the IRCS Workshop on Linguistic Databases*, pp. 105–114, Philadelphia, USA.

Joakim Nivre (2003), Theory-Supporting Treebanks, in *Proceedings of the Second Workshop on Treebanks and Linguistic Theories*, `http://w3.msi.vxu.se/~rics/TLT2003/doc/nivre.pdf`.

Adam Przepiórkowski (2008), *Powierzchniowe przetwarzanie języka polskiego*, Exit, Warsaw.

Adam Przepiórkowski (2009), A comparison of two morphosyntactic tagsets of Polish, in Violetta Koseska-Toszewa, Ludmila Dimitrova, and Roman Roszko, editors, *Representing Semantics in Digital Lexicography: Proceedings of MONDILEX Fourth Open Workshop*, pp. 138–144, Warsaw.

Adam Przepiórkowski, Rafał L. Górski, Barbara Lewandowska-Tomaszczyk, and Marek Łaziński (2008), Towards the National Corpus of Polish, in *Proceedings of the Sixth International Conference on Language Resources and Evaluation, LREC 2008*, ELRA, Marrakech.

Adam Przepiórkowski, Rafał L. Górski, Marek Łaziński, and Piotr Pęzik (2010), Recent Developments in the National Corpus of Polish, in *Proceedings of the Sixth International Conference on Language Resources and Evaluation, LREC 2010*, ELRA, Valetta, Malta.

Adam Przepiórkowski and Marcin Woliński (2003), A Flexemic Tagset for Polish, in *Proceedings of the Workshop on Morphological Processing of Slavic Languages, EACL 2003*, pp. 33–40.

Victoria Rosén, Koenraad de Smedt, and Paul Meurer (2006), Towards a Toolkit Linking Treebanking to Grammar Development, in Jan Hajič and Joakim Nivre, editors, *Proceedings of the Fifth Workshop on Treebanks and Linguistic Theories*, pp. 55–66, URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.122.2620&rep=rep1&type=pdf`.

Kiril Simov (2003), HPSG-Based Annotation Scheme for Corpora Development and Parsing Evaluation, in *Proceedings of the RANLP 2003 Conference*, pp. 432–439, Borovets, Bulgaria.

Kiril Simov, Petya Osenova, Milena Slavcheva, Sia Kolkovska, Elisaveta Balabanova, Dimitar Doikoff, Krassimira Ivanova, Alexander Simov, Er Simov, and Milen Kouylekov (2002), Building a Linguistically Interpreted Corpus of Bulgarian: the BulTreeBank, in *Proceedings of LREC 2002, Canary Islands*, pp. 1729–1736.

Marek Świdziński (1992), *Gramatyka formalna języka polskiego*, Rozprawy Uniwersytetu Warszawskiego, Wydawnictwa Uniwersytetu Warszawskiego, Warszawa.

Marek Świdziński and Marcin Woliński (2009), A new formal definition of Polish nominal phrases, in *Aspects of Natural Language Processing*, LNCS 5070, pp. 143–162, Springer.

Marek Świdziński and Marcin Woliński (to appear), Towards a Bank of Constituent Parse Trees for Polish, in *Proceedings of TSD 2010*.

Marcin Woliński (2004), *Komputerowa weryfikacja gramatyki Świdzińskiego*, Ph.D. thesis, Instytut Podstaw Informatyki PAN, Warszawa.

Marcin Woliński (2006), Jak się nie zgubić w lesie, czyli o wynikach analizy składniowej według gramatyki Świdzińskiego, *Poradnik Językowy*, 9:102–114.