Dependency treebanks: methods, annotation schemes and tools

Tuomo Kakkonen
Department of Computer Science
University of Joensuu
P.O. Box 111
FI-80101 Joensuu
tuomo.kakkonen@cs.joensuu.fi

Abstract

In this paper, current dependency-based treebanks are introduced and analyzed. The methods used for building the resources, the annotation schemes applied, and the tools used (such as POS taggers, parsers and annotation software) are discussed.

1 Introduction

Annotated data is a crucial resource for developments in computational linguistics and natural language processing. Syntactically annotated corpora, *treebanks*, are needed for developing and evaluating natural language processing applications, as well as for research in empirical linguistics. The choice of annotation type in a treebank usually boils down to two options: the linguistic resource is annotated either according to some constituent or functional structure scheme. As the name treebank suggests, these linguistic resources were first developed in the phrase-structure framework, usually represented as tree-shaped constructions. The first efforts to create such resources started around 30 years ago. The most well-known of such a treebank is the *Penn Treebank* for English (Marcus et al., 1993).

In recent years, there has been a wide interest towards functional annotation of treebanks. In particular, many dependency-based treebanks have been constructed. In addition, grammatical function annotation has been added to some constituent-type treebanks. *Dependency Grammar* formalisms stem from the work of Tesnieére (1959). In dependency grammars, only the lexical nodes are recognized, and the phrasal ones are omitted. The lexical nodes are linked with directed binary relations. The most commonly used argument for selecting the dependency format for building a treebank is that the treebank is being created for a language with a relatively free word order. Such treebanks exist *e.g.* for Basque, Czech, German and Turkish. On the other hand, dependency treebanks have been developed for languages such as English, which have been usually seen as languages that can be better represented with constituent formalism. The motivations for using dependency annotation vary from the fact that the type of structure is the one needed by many, if not most, applications to the fact that it offers a proper interface between syntactic and semantic representation. Furthermore, dependency structures can be automatically converted into phrase structures if needed (Lin, 1995; Xia and Palmer, 2000), although not always with 100% accuracy. The *TIGER Treebank* of German, a free word order language, with 50,000 sentences is an example of a treebank with both phrase structure and dependency annotations (Brants et al., 2002).

The aim of this paper is to answer the following questions about the current state-of-art in dependency treebanking:

- What kinds of texts do the treebanks consist of?
- What types of annotation schemes and formats are applied?
- What kinds of annotation methods and tools are used for creating the treebanks?
- What kinds of functions do the annotation tools for creating the treebanks have?

We start by introducing the existing dependency-based treebanks (Section 2). In Section 3, the status and state-of-art in dependency treebanking is summarized and

S. Werner (ed.), *Proceedings of the 15th NODALIDA conference, Joensuu 2005*, Ling@JoY 1, 2006, pp. 94–104
ISBN 952-458-771-8, ISSN 1796-1114. http://ling.joensuu.fi/lingjoy/

analyzed. Finally in Section 4, we conclude the findings.

2 Existing dependency treebanks

2.1 Introduction

Several kinds of resources and tools are needed for constructing a treebank: *annotation guidelines* state the conventions that guide the annotators throughout their work, a software tool is needed to aid the annotation work, and in the case of semi-automated treebank construction, a *part-of-speech (POS) tagger*, *morphological analyzer* and/or a *syntactic parser* are also needed. Building trees manually is a very slow and error-prone process. The most commonly used method for developing a treebank is a combination of automatic and manual processing, but the practical method of implementation varies considerably. There are some treebanks that have been annotated completely manually, but with taggers and parsers available to automate some of the work such a method is rarely employed in state-of-the-art treebanking.

2.2 The treebanks

2.2.1 Prague Dependency Treebank

The largest of the existing dependency treebanks (around 90,000 sentences), the *Prague Dependency Treebank* for Czech, is annotated in layered structure annotation, consisting of three levels: morphological, analytical (syntax), and tectogrammatical (semantics) (Böhmová et al., 2003). The data consist of newspaper articles on diverse topics (*e.g.* politics, sports, culture) and texts from popular science magazines, selected from the *Czech National Corpus*. There are 3,030 morphological tags in the morphological tagset (Hajič, 1998). The syntactic annotation comprises of 23 dependency types.

The annotation for the levels was done separately, by different groups of annotators. The morphological tagging was performed by two human annotators selecting the appropriate tag from a list proposed by a tagging system. Third annotator then resolved any differences between the two annotations. The syntactic annotation was at first done completely manually, only by the aid of ambiguous morphological tags and a graphical user interface. Later, some functions for automatically assigning part of the tags were implemented. After some 19,000 sentences were annotated, *Collins lexicalized stochastic parser* (Nelleke et al., 1999) was trained with the data, and was capable of assigning 80% of the dependencies correct. At that stage, the work of the annotator changed from building the trees from scratch to checking and correcting the parses assigned by the parser, except for the analytical functions, which still had to be assigned manually. The details related to the tectogrammatical level are omitted here. Figure 1 illustrates an example of morphological and analytical levels of annotation.

There are other treebank projects using the framework developed for the Prague Dependency Treebank. *Prague Arabic Dependency Treebank* (Hajič et al., 2004), consisting of around 49,000 tokens of newswire texts from *Arabic Gigaword* and *Penn Arabic Treebank*, is a treebank of Modern Standard Arabic. The *Slovene Dependency Treebank* consists of around 500 annotated sentences obtained from the *MULTEXT-East Corpus* (Erjavec, 2005b; Erjavec, 2005a).

2.2.2 TIGER Treebank

The *TIGER Treebank* of German (Brants et al., 2002) was developed based on the *NEGRA Corpus* (Skut et al., 1998) and consists of complete articles covering diverse topics collected from a German newspaper. The treebank has around 50,000 sentences. The syntactic annotation combining both phrase-structure and dependency representations is organized as follows: phrase categories are marked in non-terminals, POS information in terminals and syntactic functions in the edges. The syntactic annotation is rather simple and flat in order to reduce the amount of attachment ambiguities. An interesting feature in the treebank is that a MySQL database is used for storing the annotations, from where they can be exported into *NEGRA Export* and *TIGER-XML* file formats, which makes it usable and exchangeable with a range of tools.

The annotation tool *Annotate* with two methods, interactive and *Lexical-Functional Grammar* (LFG) parsing, was employed in creating the treebank. *LFG parsing* is a typical semi-automated annotation method, comprising of processing the input texts by a parser and a human annotator disambiguating and correcting the output. In the case of TIGER Treebank, a

```
<f cap>Do<l>do<t>RR–2———-<A>AuxP<r>1<g>7
<f num>15<l>15<t>C=————-<A>Atr <r>2<g>4
<d>.<l>.<t>Z:————-<A>AuxG<r>3<g>2
<f>května<l>květen<t>NNIS2——A—-<A>Adv<r>4<g>1
<f>budou<l>být<t>VB-P—3F-AA—<A>AuxV<r>5<g>7
<f>cestující<l>cestující<t>NNMP1——A—-<A>Sb <r>6<g>7
<f>platit<l>platit<t>Vf———A—-<A>Pred<r>7<g>0
<f>dosud<l>dosud<t>Db—————-<A>Adv<r>8<g>9
<f>platným<l>platný<t>AAIS7—-1A—-<A>Atr<r>9<g>10
<f>zpøusobem<l>způsob<t>NNIS7——A—-<A>Adv<r>10<g>7
<d>.<l>.<t>Z:————-<A>AuxK<r>11<g>0
```

Figure 1: A morphologically and analytically annotated sentence from the Prague Dependency Treebank (Böhmová et al., 2003).

broad coverage LFG parser is used, producing the constituent and functional structures for the sentences. As almost every sentence is left with unresolved ambiguities, a human annotator is needed to select the correct ones from the set of possible parses. As each sentence of the corpus has several thousands of possible LFG representations, a mechanism for automatically reducing the number of parses is applied, dropping the number of parses represented to the human annotator to 17 on average. *Interactive annotation* is also a type of semi-automated annotation, but in contrast to human post-editing, the method makes the parser and the annotator to interact. First, the parser annotates a small part of the sentence and the annotator either accepts or rejects it based on visual inspection. The process is repeated until the sentence is annotated completely.

2.2.3 Arboretum, L'Arboratoire, Arborest and Floresta Sintá(c)tica

Arboretum of Danish (Bick, 2003), *L'Arboratoire* of French and *Floresta Sintá(c)tica* of Portuguese (Afonso et al., 2002), and *Arborest* of Estonian (Bick et al., 2005) are "sibling" treebanks, Arboretum being the "oldest sister". The treebanks are hybrids with both constituent and dependency annotation organized into two separate levels. The levels share the same morphological tagset. The dependency annotation is based on the *Constraint Grammar* (CG) (Karlsson, 1990) and consists of 28 dependency types. For creating each of the four treebanks, a CG-based morphological analyzer and parser was applied. The annotation process consisted of CG parsing of the texts followed by conversion to constituent format, and manual checking of the structures.

Arboretum has around 21,600 sentences annotated with dependency tags, and of those, 12,000 sentences have also been marked with constituent structures (Bick, 2003; Bick, 2005). The annotation is in both TIGER-XML and PENN export formats. Floresta Sintá(c)tica consists of around 9,500 manually checked (version 6.8, October 15th, 2005) and around 41,000 fully automatically annotated sentences obtained from a corpus of newspaper Portuguese (Afonso et al., 2002). *Arborest* of Estonian consists of 149 sentences from newspaper articles (Bick et al., 2005). The morphosyntactic and CG-based surface syntactic annotation are obtained from an existing corpus, which is converted semi-automatically to Arboretum-style format.

2.2.4 The Dependency Treebank for Russian

The *Dependency Treebank for Russian* is based on the *Uppsala University Corpus* (Lönngren, 1993). The texts are collected from contemporary Russian prose, newspapers, and magazines (Boguslavsky et al., 2000; Boguslavsky et al., 2002). The treebank has about 12,000 annotated sentences. The annotation scheme is XML-based and compatible with *Text Encoding for Interchange* (TEI), except for some added elements. It consists of 78 syntactic relations, divided into six subgroups, such as attributive, quantitative, and coordinative. The annotation is layered, in the sense that the levels of annotation are independent and can be extracted or processed independently.

The creation of the treebank started by processing the texts with a morphological analyzer and a syntactic parser, *ETAP* (Apresjan et al., 1992), and was followed by post-editing by human annotators. Two tools are available for the annotator: a *sentence boundary markup tool* and *post-editor*. The post-editor offers the annotator functions for building, editing, and managing the annotations. The editor has a special split-and-run mode, used when the parsers fails to produce a parse or creates a parse with a high number of errors. In the mode the user can pre-chunk the sentence into smaller pieces to be input to the parser. The parsed chucks can be linked by the annotator, thus producing a full parse for the sentence. The tool also provides the annotator with the possibility to mark the annotation of any word or sentence as doubtful, in order to remind at the need for a later revision.

2.2.5 Alpino

The *Alpino Treebank* of Dutch, consisting of 6,000 sentences, is targeted mainly at parser evaluation and comprises of newspaper articles (van der Beek et al., 2002). The annotation scheme is taken from the *CGN Corpus* of spoken Dutch (Oostdijk, 2000) and the annotation guidelines are based on the TIGER Treebank's guidelines.

The annotation process in the Alpino Treebank starts with applying a parser based on *Head-Driven Phrase Structure Grammar* (HPSG) (Pollard and Sag, 1994) and is followed by a manual selection of the correct parse trees. An interactive lexical analyzer and a constituent marker tools are employed to restrict the number of possible parses. The *interactive lexical analyzer* tool lets the user to mark each word in a sentence belonging to correct, good, or bad categories. 'Correct' denotes that the parse includes the lexical entry in question, 'good' that the parse may include the entry, and 'bad' that the entry is incorrect. The parser uses this manually reduced set of entries, thus generating a smaller set of possible parses. With the constituent marker tool, the annotator can mark constituents and their types to sentences, thus aiding the parser.

The selection of the correct parse is done by the help of a *parse selection tool*, which calculates *maximal discriminants* to help the annotator. There are three types of discriminants. Maximal discriminants are sets of shortest dependency paths encoding differences between parses, lexical discriminants represent ambiguities resulting from lexical analysis, and constituent discriminants group words to constituents without specifying the type of the constituent. The annotator marks each of the maximal discriminants as good or bad, and the tool narrows down the number of possible parses based on the information. If the parse resulting from the selection is not correct, it can be edited by a parse editor tool.

2.2.6 The Danish Dependency Treebank

The annotation of the *Danish Dependency Treebank* is based on *Discountinuous Grammar*, which is a formalism closely related to *Word Grammar* (Kromann, 2003). The treebank consists of 5,540 sentences covering a wide range of topics. The morphosyntactic annotation is obtained from the *PAROLE Corpus* (Keson and Norling-Christensen, 2005), thus no morphological analyzer or POS tagger is applied. The dependency links are marked manually by using a command-line interface with a graphical parse view. A parser for automatically assigning the dependency links is under development.

2.2.7 METU-Sabanci Turkish Treebank

Morphologically and syntactically annotated *Turkish Treebank* consists of 5,000 sentences obtained from the *METU Turkish Corpus* (Atalay et al., 2003) covering 16 main genres of present-day written Turkish (Oflazer et al., 2003). The annotation is presented in a format that is in conformance with the *XML-based Corpus Encoding Standard* (XCES) format (Anne and Romary, 2003). Due to morphological complexity of Turkish, morphological information is not encoded with a fixed set of tags, but as sequences of *inflectional groups* (IGs). An IG is a sequence of inflectional morphemes, divided by derivation boundaries. The dependencies between IGs are annotated with the following 10 link types: subject, object, modifier, possessor, classifier, determiner, dative adjunct, locative adjunct, ablative adjunct, and instrumental adjunct. Figure 2 illustrates a sample annotated sentence from the treebank.

The annotation, directed by the guidelines, is done in a semi-automated fashion, although relatively lot of manual work remains. First, a morphological analyzer based on the two-level morphology model (Oflazer, 1994) is ap-

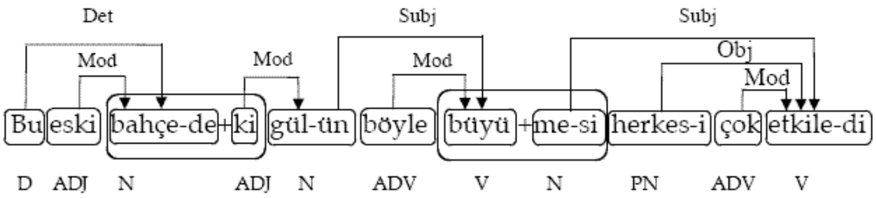

Figure 2: A sample sentence from the METU-Sabanci Treebank (Oflazer et al., 2003).

plied to the texts. The morphologically analyzed and preprocessed text is input to an annotation tool. The tagging process requires two steps: morphological disambiguation and dependency tagging. The annotator selects the correct tag from the list of tags proposed by the morphological analyzer. After the whole sentence has been disambiguated, dependency links are specified manually. The annotators can also add notes and modify the list of dependency link types.

2.2.8 The Basque Dependency Treebank

The *Basque Dependency Treebank* (Aduriz and al., 2003) consists of 3,000 manually annotated sentences from newspaper articles. The syntactic tags are organized as a hierarchy. The annotation is done by aid of an annotation tool, with tree visualization and automatic tag syntax checking capabilities.

2.2.9 The Turin University Treebank

The *Turin University Treebank* for Italian consisting of 1,500 sentences is divided into four sub-corpora (Lesmo et al., 2002; Bosco, 2000; Bosco and Lombardo, 2003). The majority of texts is from civil law code and newspaper articles. The annotation format is based on the *Augmented Relational Structure* (ARS). The POS tagset consists of 16 categories and 51 subcategories. There are around 200 dependency types, organized as a taxonomy of five levels. The scheme provides the annotator with the possibility of marking a relation as *underspecified* if a correct relation type cannot be determined.

The annotation process consists of automatic tokenization, morphological analysis and POS disambiguation, followed by syntactic parsing (Lesmo et al., 2002). The annotator can interact with the parser through a graphical interface, in a similar way to the interactive method in the TIGER Treebank. The annotator can either accept or reject the suggested tags for each word in the sentence after which the parser proceeds to the next word (Bosco, 2000).

2.2.10 The Dependency Treebank of English

The *Dependency Treebank of English* consists of dialogues between a travel agent and customers (Rambow et al., 2002), and is the only dependency treebank with spoken language annotation. The treebank has about 13,000 words. The annotation is a direct representation of lexical predicate-argument structure, thus arguments and adjuncts are dependents of their predicates and all function words are attached to their lexical heads. The annotation is done at a single, syntactic level, without surface representation for surface syntax, the aim being to keep the annotation process as simple as possible. Figure 3 shows an example of an annotated sentence.

The trained annotators have access to an on-line manual and work off the transcribed speech without access to the speech files. The dialogs are parsed with a dependency parser, the *Supertagger and Lightweight Dependency Analyzer* (Bangalore and Joshi, 1999). The annotators correct the output of the parser using a graphical tool, the one developed by Prague Dependency Treebank project. In addition to the standard tag editing options, annotators can add comments. After the editing is done, the sentence is automatically checked for inconsistencies, such as the difference in surface and deep roles or prepositions missing objects *etc.*

2.2.11 DEPBANK

As the name suggests, the *PARC 700 Dependency Bank* (DEPBANK) (King et al., 2003) consists of 700 annotated sentences from the *Penn Wall Street Journal Treebank* (Marcus et al., 1994). There are 19 grammatical relation types (*e.g.* subject, object, modifier) and 37

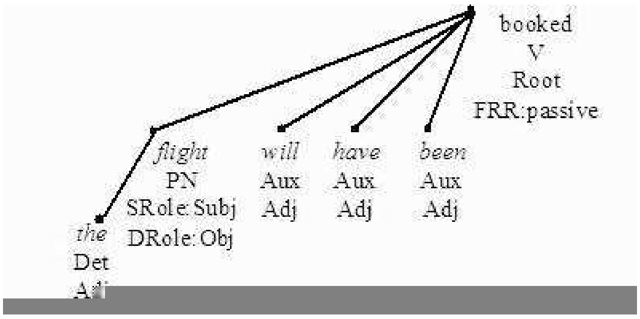

Figure 3: The sentence "The flight will have been booked" from the English treebank (Rambow et al., 2002). The words are marked with the word form (first line), the POS (second line), and the surface role (third line). In addition, node 'flight' is marked with a deep role (DRole) and the root node as passive in the FRR feature, not set in any other nodes.

feature types (*e.g.* number (pl/sg), passive (+/-), tense (future/past/present)) in the annotation scheme.

The annotation process is semi-automatic, consisting of parsing by broad-coverage LFG, converting the parses to the DEPBANK format and manually checking and correcting the resulting structures. The annotations are checked by a tool that looks *e.g.* for the correctness of header information and the syntax of the annotation, and inconsistencies in feature names. The checking tool helps in two different ways: first, when the annotator makes corrections to the parsed structure, it makes sure that no errors were added, and second, the tool can detect erroneous parses and note that to the annotator.

3 Analysis

Table 1 summarizes some key properties of the existing dependency treebanks. The size of the treebanks is usually quite limited, ranging from few hundreds to 90,000 sentences. This is partly due to the fact that even the most long-lived of the dependency treebank projects, the Prague Dependency Treebank, was started less than 10 years ago. The treebank producers have in most cases aimed at creating a multi-purpose resource for evaluating and developing NLP systems and for studies in theoretical linguistics. Some are built for specific purposes, *e.g.* the Alpino Treebank of Dutch is mainly for parser evaluation. Most of the dependency treebanks consist of written text; to our knowledge there is only one that is based on a collection of spoken utterances. The written texts are most commonly obtained from newspaper articles, and in the cases of *e.g.* Czech, German, Russian, Turkish, Danish, and Dutch treebanks from an existing corpus. Annotation usually consists of POS and morphological levels accompanied by dependency-based syntactic annotation. In the case of the Prague Dependency Treebank a higher, semantic layer of annotation is also included.

The definition of the annotation schema is always a trade-off between the accuracy of the representation, data coverage and cost of treebank development (Bosco, 2000; Bosco and Lombardo, 2003). The selection of the tagsets for annotation is critical. Using a large variety of tags provides a high accuracy and specialization in the description, but makes the annotators' work even more time-consuming. In addition, for some applications, such as training of statistical parsers, highly specific annotation easily leads into sparsity problem. On the other hand, if annotation is done at a highly general level the annotation process is faster, but naturally lot of information is lost. The TUT and Basque treebanks try to tackle the problem by organizing the set of grammatical relations into hierarchical taxonomy. Also the choice of type of application for the treebank may affect the annotation choices. A treebank for evaluation allows for some remaining ambiguities but no errors, while the opposite may be true for a treebank for training (Abeillé, 2003). In annotation consisting of multiple levels clear separation between the levels is a concern. The format of the annotation is also directed by the specific language that the treebank is being developed for. The format must be suited for representing the structures of the language. For

Table 1: Comparison of dependency treebanks. (*Due to limited number of pages not all the treebanks in the Arboretum "family" are included in the table. **Information of number of utterances was not available. M=manual, SA=semi-automatic, TB=treebank)

Name	**Language**	**Genre**	**Size (sent.)**	**Annotation methods**	**Autom. tools**	**Supported formats**
Prague Dep. TB	Czech	Newsp., science mag.	90,000	M/SA	Lexicalized stochastic parser (Collins)	FS, CSTS SGML, Annotation Graphs XML
TIGER TB	German	Newsp.	50,000	Post-editing & interactive	Probabilistic/ LFG parser	TIGER-XML & NEGRA export
*Arboretum & co.**	4 lang.	Mostly newsp.	21,600 (Ar) 9,500 (Flo)	Dep. to const. conversion, M checking	CG-based parser for each language	TIGER-XML & PENN export (Ar.)
Dep. TB for Russian	Russian	Fiction, newsp., scientific	12,000	SA	Morph. analyzer & a parser	XML-based TEI-compatibe
Alpino	Dutch	Newsp.	6,000	M disambig. aided by parse selection tool	HPSG-based Alpino parser	Own XML-based
Danish Dep. TB	Danish	Range of topics & genres	5,540	Morphosyn. annotation obtained from a corpus, M dep. marking	-	PAROLE-DK with additions, TIGER-XML
METU-Sabanci TB	Turkish	16 genres	5,000	M disambiguation & M dependency marking	Morph. analyzer based on XEROX FST	XML-based XCES compatible
Basque TB	Basque	Newsp.	3,000	M, automatic checking	-	XML-based TEI-compatible
TUT	Italian	Mainly newsp. & civil law	1,500	M checking of parser & morph. analyzer output	Morph. analyzer, rule-based tagger and a parser	Own ASCII-based
Dep. TB of English	English	Spoken, travel agent dial.	13,000 words **	M correction of parser output & autom. checking of inconsistencies	Supertagger & Lightweight Dep. Analyzer	FS
DEP-BANK	English	Financial newsp.	700	M checking & correction, autom. consistency checking	LFG parser, checking tool	Own ASCII-based

example, in the METU-Sabanci Treebank a special type of morphological annotation scheme was introduced due to the complexity of Turkish morphology.

Semi-automated creation combining parsing and human checker is the state-of-art annotation method. None of the dependency treebanks are created completely manually; at least an annotation tool capable of visualizing the structures is used by each of the projects. Obviously, the reason that there aren't any fully automatically created dependency treebanks is the fact there are no parsers of free text capable of producing error-free parses.

The most common way of combining the human and machine labor is to let the human work as a post-checker of the parser's output. Albeit most straight-forward to implement, the method has some pitfalls. First, starting annotation with parsing can lead to high number of unresolved ambiguities, making the selection of the correct parse a time-consuming task. Thus, a parser applied for treebank building should perform at least some disambiguation to ease the burden of annotators. Second, the work of post-checker is mechanic and there is a risk that the checker just accept the parser's suggestions, without a rigorous inspection. A solution followed *e.g.* by the both treebanks for English and the Basque treebank is to apply a post-checking tool to the created structures before accepting them. Some variants of semi-automated annotation exist: the TIGER, TUT, Alpino, and the Russian Treebanks apply a method where the parser and the annotator can interact. The advantage of the method is that when the errors by the parser are corrected by the human at the lower levels, they do not multiply into the higher levels, thus making it more probable that the parser produces a correct parse. In some annotation tools, such as the tools of the Russian, the English Dependency treebanks, the annotator is provided with the possibility of adding comments to annotation, easing the further inspection of doubtful structures. In the annotation tool of the TUT Treebank, a special type relation can be assigned to mark doubtful annotations.

Although more collaboration has emerged between treebank projects in recent years, the main problem with current treebanks in regards to their use and distribution is the fact that instead of reusing existing formats, new ones have been developed. Furthermore, the schemes have often been designed from theory and even application-specific viewpoints, and consequently, undermine the possibility for reuse. Considering the high costs of treebank development (for example in the case of the Prague Dependency Treebank estimated USD600,000 (Böhmová et al., 2003)), reusability of tools and formats should have a high priority. In addition to the difficulties for reuse, creating a treebank-specific representation format requires developing a new set of tools for creating, maintaining and searching the treebank. Yet, the existence of exchange formats such as XCES (Anne and Romary, 2003) and TIGER-XML (Mengel and Lezius, 2000) would allow multipurpose tools to be created and used.

4 Conclusion

We have introduced the state-of-art in dependency treebanking and discussed the main characteristics of current treebanks. The findings reported in the paper will be used in designing and constructing an annotation tool for dependency treebanks and constructing a treebank for Finnish for syntactic parser evaluation purposes. The choice of dependency format for a treebank for evaluating syntactic parser of Finnish is self-evident, Finnish being a language with relatively free word order and all parsers for the language working in the dependency-framework. The annotation format will be some of the existing XML-based formats, allowing existing tools to be applied for searching and editing the treebank.

The findings reported in this paper indicate that the following key properties must be implemented into the annotation tool for creating the treebank for Finnish:

- An interface to a morphological analyzer and parser for constructing the initial trees. Several parsers can be applied in parallel to offer the annotator a possibility to compare the outputs.
- Support for an existing XML annotation format. Using an existing format will make the system more reusable. XML-based formats offer good syntax-checking capabilities.
- An inconsistency checker. The annotated

sentences to be saved will be checked against errors in tags and annotation format. In addition to XML-based validation of the syntax of the annotation, the inconsistency checker will inform the annotator about several other types of mistakes. The POS and morphological tags will be checked to find any mismatching combinations. A missing main verb, a fragmented, incomplete parse etc. will be indicated to the user.

- A comment tool. The annotator will be able to add comments to the annotations to aid later revision.
- Menu-based tagging. In order to minimize errors, instead of typing the tags, the annotator will only be able to set tags by selecting them from predefined lists.

5 Acknowledgments

The research reported in this work has been supported by the European Union under a Marie Curie Host Fellowship for Early Stage Researchers Training at MULTILINGUA, Bergen, Norway and MirrorWolf project funded by the National Technology Agency of Finland (TEKES).

References

Anne Abeillé. 2003. Introduction. In Anne Abeillé, editor, *Treebanks: Building and Using Syntactically Annotated Corpora*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Itziar Aduriz and al. 2003. Construction of a Basque dependency treebank. In *Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories*, Växjö, Sweden.

Susana Afonso, Eckhard Bick, Renato Haber, and Diana Santos. 2002. "Floresta Sintá(c)tica": a treebank for Portuguese. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, Las Palmas, Gran Canaria, Spain.

Nancy Anne and Laurent Romary. 2003. Encoding syntactic annotation. In Anne Abeillé, editor, *Treebanks: Building and Using Syntactically Annotated Corpora*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Jurij Apresjan, Igor Boguslavsky, Leonid Iomdin, Alexandre Lazursku, Vladimir Sannikov, and Leonid Tsinman. 1992. ETAP-2: the linguistics of a machine translation system. *META*, 37(1):97–112.

Nart B. Atalay, Kermal Oflazer, and Bilge Say. 2003. The annotation process in the Turkish treebank. In *Proceedings of the EACL Workshop on Linguistically Interpreted Corpora*, Budapest, Hungary.

Srinivas Bangalore and Aravind K. Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics*, 25(2):237–265.

Eckhard Bick, Heli Uibo, and Kaili Müürisep. 2005. Arborest - a growing treebank of Estonian. In Henrik Holmboe, editor, *Nordisk Sprogteknologi 2005. Nordic Language Technology 2005.* Museum Tusculanums Forlag, Copenhagen, Denmark.

Eckhard Bick. 2003. Arboretum, a hybrid treebank for Danish. In *Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories*, Växjö, Sweden.

Eckhard Bick. 2005. Current status of Arboretum treebank. Electronic mail communication, October 12th, 2005.

Igor Boguslavsky, Svetlana Grigorieva, Nikolai Grigoriev, Leonid Kreidlin, and Nadezhda Frid. 2000. Dependency treebank for Russian: Concept, tools, types of information. In *Proceedings of the 18th International Conference on Computational Linguistics*, Saarbrücken, Germany.

Igor Boguslavsky, Ivan Chardin, Svetlana Grigorieva, Nikolai Grigoriev, Leonid Iomdin, Leonid Kreidlin, and Nadezhda Frid. 2002. Development of a dependency treebank for Russian and its possible applications in NLP. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, Las Palmas, Gran Canaria, Spain.

Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. 2003. The Prague dependency treebank: Three-level annotation scenario. In Anne Abeillé, editor, *Treebanks: Building and Using Syntactically Annotated Corpora*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Cristina Bosco and Vincenzo Lombardo. 2003. A relation-based schema for treebank annotation. In *Proceedings of the Advances in Artificial Intelligence, 8th Congress of the Italian Association for Artificial Intelligence*, Pisa, Italy.

Cristina Bosco. 2000. An annotation schema for an Italian treebank. In *Proceedings of the Student Session, 12th European Summer School in Logic, Language and Information*, Birmingham, UK.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The Tiger Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, Sozopol, Bulgaria.

Tomaž Erjavec. 2005a. Current status of the Slovene dependency treebank. Electronic mail communication, September 29th, 2005.

Tomaž Erjavec. 2005b. Slovene dependency treebank. http://nl.ijs.si/sdt/. (Accessed November 9th, 2005).

Jan Hajič, Otakar Smrž, Petr Zemáánek, Jan Šnaidauf, and Emanuel Beška. 2004. Prague Arabic dependency treebank: Development in data and tools. In *Proceedings of the NEMLAR International Conference on Arabic Language Resources and Tools*, Cairo, Egypt.

Jan Hajič. 1998. Building a syntactically annotated corpus: The Prague dependency treebank. In Eva Hajičová, editor, *Issues of Valency and Meaning. Studies in Honor of Jarmila Panevová*, pages 12–19. Charles University Press, Prague, Czech Republic.

Fred Karlsson. 1990. Constraint grammar as a framework for parsing running text. In *Proceedings of the 13th Conference on Computational Linguistics - Volume 3*, Helsinki, Finland.

Britt-Katrin Keson and Ole Norling-Christensen. 2005. PAROLE-DK. Det Danske Sprog- litteraturselskab. http://korpus.dsl.dk/e-resurser/parole-korpus.php. (Accessed October 7th, 2005).

Tracy H. King, Richard Crouch, Stefan Riezler, Mary Dalrymple, and Ronald M. Kaplan. 2003. The PARC 700 dependency bank. In *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora, the 10th Conference of the EACL*, Budapest, Hungary.

Matthias T. Kromann. 2003. The Danish dependency treebank and the DTAG treebank tool. In *Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories*, Växjö, Sweden.

Leonardo Lesmo, Vincenzo Lombardo, and Cristina Bosco. 2002. Treebank development: the TUT approach. In *Proceedings of the International Conference on Natural Language Processing*, Mumbai, India.

Dekang Lin. 1995. A dependency-based method for evaluating broad-coverage parsers. In *The 1995 International Joint Conference on AI*, Montreal, Canada.

Lennart Lönngren. 1993. Chastotnyj slovar' sovremennogo russkogo jazyka. (A frequency dictionary of modern Russian). In *Acta Universitatis Upsaliensis, Studia Slavica Upsaliensia 32*, Uppsala, Sweden.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn treebank: Annotating predicate argument structure. In *Proceedings of the ARPA Human Language Technology Workshop*, Princeton, New Jersey, USA.

Andreas Mengel and Wolfgang Lezius. 2000. An XML-based representation format for syntactically annotated corpora. In *Proceedings of the International Conference on Language Resources and Evaluation*, Athens, Greece.

Michael Nelleke, Jan Hajič, Eric Brill, Lance Ramshaw, and Christoph Tillmann. 1999. A statistical parser of Czech. In *Proceedings of 37th Annual Meeting of the Association for Computational Linguistics*, College Park, Maryland, USA.

Kermal Oflazer, Bilge Say, Dilek Zeynep Hakkani-Tür, and Gökhan Tür. 2003. Building a Turkish treebank. In Anne Abeillé, editor, *Treebanks: Building and Using Syntactically Annotated Corpora*. Kluwer Academic Publishers.

Kemal Oflazer. 1994. Two-level description of Turkish morphology. *Literary and Linguistic Computing*, 9(2):472–472.

Nelleke Oostdijk. 2000. The Spoken Dutch Corpus: Overview and first evaluation. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation*, Athens, Greece.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press and CSLI Publications, Chicago, Illinois, USA.

Owen Rambow, Cassandre Creswell, Rachel Szekely, Harriet Taber, and Marilyn Walker.

2002. A dependency treebank for English. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, Las Palmas, Gran Canaria, Spain.

Wojciech Skut, Thorsten Brants, Birgitte Krenn, and Hans Uszkoreit. 1998. A linguistically interpreted corpus of German newspaper text. In *Proceedings of the 10th European Summer School in Logic, Language and Information*, Saarbrücken, Germany.

Lucien Tesnieére. 1959. *Éléments de syntaxe structurale*. Klincksiek, Paris, France.

Leonoor van der Beek, Gosse Bouma, Robert Malouf, and Gertjan van Noord. 2002. The Alpino dependency treebank. In Mariet Theune, Anton Nijholt, and Hendri Hondorp, editors, *Computational Linguistics in the Netherlands 2001. Selected Papers from the Twelfth CLIN Meeting*. Rodopi, Amsterdam, The Netherlands.

Fei Xia and Martha Palmer. 2000. Converting dependency structures to phrase structures. In *Proceedings of the First International Conference on Human Language Technology Research*, San Diego, California, USA.