The Szeged Treebank Project

Dóra Csendes, János Csirik, András Kocsor
{dcsendes, csirik, kocsor@inf.u-szeged.hu}
University of Szeged, Department of Informatics,
Human Language Technology Group

Introduction

The major aim of the Szeged Treebank project was to create a high-quality database of syntactic structures for Hungarian that can serve as a golden standard to further research in linguistics and computational language processing. The treebank currently contains full syntactic parsing of about 82,000 sentences (1.2 million words), which is the result of accurate manual annotation.

Inspired by the research results of the Penn Treebank [6] and several other treebank projects [1,2,3,5,7], our research group set out to create a golden standard treebank for Hungarian, containing reliable syntactic annotation of texts. Project work contained the selection and adjustment of the theory used for syntactic analysis, the design of the annotation methodology, the adaptation of the available tag-sets to Hungarian, automated pre-processing, manual validation and correction, and experiments with machine learning methods for automated parsing. The proposed poster is to presents an overview of the Szeged Treebank initiative and its results to date.

Ideally, the treebank should contain samples of all the syntactic structures of the language, therefore, it serves as a reference for future corpus and treebank developments, grammar extraction and other linguistic research. It also serves as a reliable test suite for different NLP applications, as well as a basis for the development of computational methods for both shallow and deep syntactic parsing, and information extraction. Well-defined methods or elaborate theoretical foundations for the automated syntactic analysis of Hungarian texts were lacking at the start of the project. For this reason, novelty of the project work lies in the design of a practical approach for syntactic annotation of Hungarian natural language sentences.

1. Preliminaries

The compilation of a golden standard textual database for Hungarian language was an extensive and carefully planned work with roots going back to the “MULTEXT-EAST” project for Central and Eastern-European languages[1]. The resulting Szeged Corpus is a manually annotated natural language database comprising 1.2 million word entries (with 145,000 different word forms) and an additional 225,000 punctuation marks [4]. It is a thematically representative database containing texts from six different genres, namely: fiction, newspaper articles, computation-related scientific texts, short essays of 14-16-year-old students, legal texts, and short business news. Language processing of the Szeged Corpus includes morphological analysis, POS tagging and shallow syntactic parsing. Shallow parsing went as far as marking bottom-level NP structures, and clause annotation.

2. Theoretical background

Since no syntactic annotation schemes were available for Hungarian, the major challenge of the Szeged Treebank project was to adapt the theoretical foundations of Hungarian syntax to a more practical syntactic annotation methodology. When designing the methodology, researchers aimed to (i) demonstrate the varieties of Hungarian syntactic patterns exhaustively; (ii) stay in correlation with the newest linguistic theories[2]; (iii) create an annotation scheme that can be used extensively in later research activities and in computer assisted practical solutions. Research results showed that the most promising theoretical frame for the definition of the annotation scheme would be generative syntax in combination with certain dependency formalism, (the latter being considered more suitable for languages with free word order). The created annotation scheme allows for the description of nodes with complex labels contain morphological and syntactic description of the sentence components in the form of attributes.

In building a syntactic tree, the initial step is the (re)creation of the deep sentence structure. In a deep structure of a Hungarian sentence, it is always the verb that stands in the first position and it is followed by its arguments.

[1] Erjavec, T., Monachini, M.: *Specification and Notation for Lexicon Encoding,* Copernicus Project 106 „MULTEX-EAST”, Work Package 1 – Task 1.1, Deliverable D1.1F (1997)

[2] References: É. Kiss K., Kiefer F., Siptár J.: *Új magyar nyelvtan*, Osiris Kiadó, Bp., 1999.; Alberti G., Medve A.: *Generatív grammatikai gyakorlókönyv I-II.*, Janus/Books, Bp., 2002.; Kiefer F., ed.: *Strukturális magyar nyelvtan I. Mondattan*, Akadémiai Kiadó, Bp., 1992.

Since Hungarian has a relatively free word order, arguments of the verb can move anywhere in the sentence occupying so-called functional positions. Naturally, by moving certain arguments, the meaning of the sentence is likely to change accordingly. Arguments that moved somewhere else, leave traces in their original position, which are indexed to their newly occupied position (see Figure 1.). When applying this theory to the Szeged Treebank, we decided not to keep the traces in the treebank, instead, we added a new NODE label within the verb phrase and described the given argument with attributes. The resulting syntactic trees do not appear in the form of a tree, but as bracketed structures using XML format, (however, the transformation into a tree is always possible). The first figure shows the original tree with the argument traces, while the second one illustrates our XML representation of the same sentence.

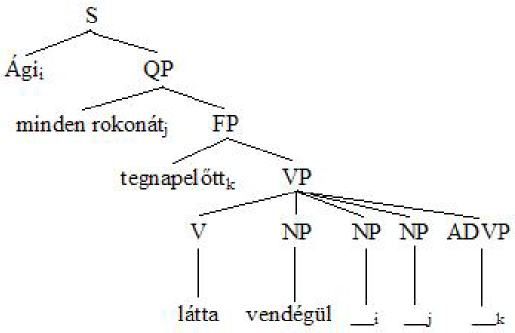

Figure 1.

```
<CP id="file.1.1">
        <NP id="file.1.2"> Ági </NP>
        <NP id="file.1.3">
          <ADJP> minden </ADJP>
          rokonát
        </NP>
        <ADVP id="file.1.4"> tegnapelőtt </ADVP>
        <V_ id="file.1.5">
          <V0> látta </V0>
          <CHILDREN>
                    <NODE idref="file.1.2" type="NP" role="NOM"> </NODE>
                    <NODE idref="file.1.3" type="NP" role="ACC"> </NODE>
                    <NODE idref="file.1.4" type="ADVP" role="TLOCY"> </NODE>
                    <NODE idref="file.1.6" type="NP" role="ESS"> </NODE>
          </CHILDREN>
        </V_>
        <NP id="file.1.6"> vendégül </NP>
        <c> . </c>
</CP>
```

Figure 2.

3. Annotation of the Szeged Treebank

Similarly to the majority of annotation projects, the Szeged Treebank also follows the Penn Treebank approach, which distinguishes an automatic annotation step followed by manual validation and correction. The tag-set used in the project shows correlation with many other internationally accepted syntactic tag-sets, see below:

ADJP:	adjectival phrases
ADVP:	adverbial phrases, adverbial adjectives, postpositional personal pronouns
c:	punctuation mark
C0:	conjunctions
CP:	clauses (also for marking sentences)
INF_:	infinitives (INF0, CHILDREN, NODE)
NEG:	negation
NP:	noun phrases (groups with noun or predicative adjective or inflected personal pronouns as head)
PA_:	adverbial participles (PA0, CHILDREN, NODE)

PP:	postpositional phrases
PREVERB:	preverbs
V_:	verb (V0, CHILDREN, NODE)
XP:	any circumstantial or parenthetic clause that is not a direct part of the sentence

Attributes of a node may contain information about the node's type (e.g., NP, ADVP, etc.), and its morpho-semantic role in the sentence (e.g., nominative, instrumental, inessive, terminative, locative, etc.) also to be seen in Figure 2.

Automatic pre-parsing of the sentences was completed with the help of the CLaRK[3] program, in which syntactic rules have been defined by linguistic experts for the recognition of NPs. The basic mechanism of CLaRK for linguistic processing of text corpora is a cascaded regular grammar processor. A remarkable ~70% accuracy was already achieved in the pre-parsing phase, due to the definition of expert rules of high efficiency. For the pre-parsing of all other structures (ADJP, ADVP, etc.), we developed our own tool, which applies manually defined simple grammatical rules for the automated pre-annotation of sentences.

Manual validation and correction of the syntactic structures and their attributes was performed by a group of linguist especially trained for this task. They used a locally developed editor for the task and worked 24 person-months on the project.

4. Training and testing machine learning algorithms for full syntactic parsing

Research groups studying the structure of Hungarian sentences have made a great effort to produce a consistent and extensive syntax rule system, yet these are not or just partially adapted to practical, computer related purposes so far. This implied that there is a strong need for the development of a technology that would be able to divide a Hungarian sentence into syntactical segments, recognize their structure, and based on this recognition, would assign an annotated tree representation to each sentence. The main goal, therefore, was to develop a generally applicable syntactic parser for Hungarian based on the Szeged Treebank annotations. Different learning methods have been studied, based on which a parser was developed taking into consideration the specific features of Hungarian language.

For training and testing the parsers, we used a set of 9600 sentences divided into 10 sections for ten-fold cross validation. The input of the parsers was morphologically analysed text and the output was bracketed syntactically analysed sentences. Parsing rules were retrieved from the annotated Szeged Corpus and were combined with manually defined ones. Tree patterns were previously defined by the following method: a pre-processor divided the sentence into comprehensive structures along verbs, conjunctions, punctuation marks, and the words in between the dividing elements formed the NP trees. By this method, ~300 different tree patterns were identified based on the sentences of the evaluation domain. The table below shows average results of the ten-fold cross validation test performed by the developed parser for the recognition of NPs.

Categories of recognition	**Precision**	**Recall**	**Fβ=1**
Complete NP structures	81.28%	87.43%	84.24%
Boundaries (first and last elements) of NP structures	88.31%	92.08%	90.15%
NP structures (depth<=2)	86.02%	89.72%	87.83%
NP structures (depth>2)	74.71%	78.19%	76.41%

Table 1. NP recognition results

In the case of full syntactic parsing, we aimed at the recognition of shorter multi-level tree structures, incl. ADJPs, ADVPs, PAs, etc. The training resulted in ~1500 different tree patterns where the leaves contain detailed morphological and morpho-semantic information about the component. We have experimented with different learning algorithms, namely: the rule-based C4.5, the numeric SVM, and the self-developed PGS logic algorithms. Test results for full parsing of short trees can be seen in the Table 2.

In general, the SVM algorithm has some advantage over the other methods (best results are bolded). A lack of advantage can be best traced in the drop of precision results by SVM, and it also has to be noted that with respect to the F-values, there is no significant difference between the methods. Taking the values of average and variance into consideration, SVM is less sensible to the minor modifications of the database than the other algorithms.

[3] The CLaRK system was developed by Kiril Simov at the Bulgarian Academy of Sciences in the framework of the BulTreeBank project (http://www.bultreebank.org).

A	baseline	C4.5	PGS	SVM
Accuracy	57,70%	85,99%	84,85%	**86,24%**
	1,11%	*2,75%*	*2,71%*	***0,35%***
Precision	-	**85,17%**	81,01%	81,88%
	-	*2,54%*	*2,91%*	***0,19%***
Recall	-	75,08%	76,14%	**76,82%**
	-	*6,22%*	*6,41%*	***0,57%***
F-value	-	**79,75%**	78,45%	79,27%
	-	*4,59%*	*4,69%*	***0,26%***

Table 2. Recognition results for full syntactic structures

The results illustrated above are only preliminary ones, and can be considered as base-line results in syntactic parsing of Hungarian sentences. It must be admitted that better results are already available for other languages (cf. results of the Link, NLTK, Stanford Parser, Apple Pie parsers), but due to the fact that this is a fresh initiative for Hungarian, and that the number of tree patterns is much higher than for other languages, results can be considered promising. Further improvements in this field are the nearest future plan of the group.

References

1. Abeillé, A., Clément, L., Toussenel, F.: *Building a Treebank for French* in A. Abeillé (ed) *Treebank:. Building and Using Parsed Corpora*, Kluwer Academic Publishers, pp 165-187 (2003)
2. Bond, F., Sanae F., Chikara H., Kaname K., Shigeko N., Nichols, E., Akira O., Takaaki T., Shigeaki A.: *The Hinoki Treebank: A Treebank for Text Understanding* in Proceedings of the IJCNLP 2004, Hainan Island, China and in LNCS vol. 3248 (2004)
3. Brants, S., Dipper, S., Hansen, S., Lezius, W. and Smith, G.: *The TIGER Treebank* in Proceedings of the Workshop on Treebanks and Linguistic Theories (TLT 2002), Sozopol, Bulgaria (2002)
4. Csendes, D., Csirik, J., Gyimóthy, T.: *The Szeged Corpus: A POS tagged and Syntactically Annotated Hungarian Natural Language Corpus* in Proceedings of TSD 2004, Brno, Czech Republic and LNAI vol. 3206 (2004)
5. Hajic, J.: *Building a Syntactically Annotated Corpus: The Prague Dependency Treebank* in Issues of Valency and Meaning, pp. 106-132, Charles University Press, Prague (1999)
6. Marcus, M., Santorini, B., Marcinkiewicz, M.: *Building a large annotated corpus of English: the Penn Treebank* in Computational Linguistics, vol. 19 (1993)
7. Simov, K., Simov, A., Kouylekov, M., Ivanova, K., Grigorov, I., Ganev, H.: *Development of Corpora within the CLaRK System: The BulTreeBank Project Experience* in Proceedings of the Demo Sessions of EACL'03, pp. 243-246, Budapest, Hungary (2003)