Applying Computer Technologies to the Georgian Language:
From a Treebank to a Syntactic Parser

Thomas Hanneforth (Potsdam University, Potsdam, Germany), Oleg Kapanadze (Tbilisi State University, Tbilisi, Georgia), Gideon Kotzé (University of South Africa, Pretoria, South Africa)

1. Introduction

A large part of the methodology for natural language processing (NLP) has been developed for English which is known as a strongly configurational language. Hence, nearly all the syntactic information needed by any NLP application for English can be obtained by configurational analysis. At the other end of the configurational spectrum are the languages with rich derivational and inflectional morphology, such as Georgian that has very little fixed structure on the sentence level. These languages for *morphologically rich and less-configurational* features are referred to as **MR&LC** [1].

There are a multitude of academic grammars and dictionaries developed for the Georgian language. However, this does not mean that there is sufficient support for computational applications involving Georgian, as these resources are not suited for NLP needs.

The proposed presentation will feature issues concerned with the development of a crucial NLP resource — a syntactic parser for the Georgian language.

2. Treebanking in NLP

In the last decade there has been an increasing interest in the construction of syntactically annotated corpora, commonly called *treebanks.* A *treebank* is a parsed corpus in which sentences are annotated with syntactic structure. They are skeletal parses showing syntactic information — a ***bank*** of linguistic ***trees***. Syntactic structure is commonly represented as ***a tree structure*** (in mathematical terms: ***an oriented graph*)**, hence, the name ***treebank*.**

Treebanks have become valuable resources as repositories for linguistic research, since corpus-based methods became useful in multilingual technology playing an important role in empirical language studies. They can be used in *contrastive studies* and *translation science,* in *corpus linguistics* for studying syntactic phenomena, in *computational linguistics* as evaluation corpora for different human language technology systems or for training and testing *parsers*, as well as for a database for *translation memory* systems.

Treebanks can be created completely manually or semi-automatically, where a parser assigns some syntactic structure to a text that is then checked by linguists and, if necessary, corrected. Treebanks are often created on top of a corpus that has already been annotated with part-of-speech tags. The annotation can vary from constituent to dependency or tecto-grammatical structures. Additionally, treebanks are sometimes enhanced with semantic or other linguistic information.

Some treebanks follow a specific linguistic theory (e.g. the Bulgarian language follows HPSG), but most try to be less theory-specific. However, two main groups can be distinguished: treebanks that annotate *phrase structure* (the *Penn Treebank* for Arabic, English and Chinese) and those that annotate *dependency structure* (the Prague Dependency Treebank for the Czech language).

3. Creating a Georgian Treebank and a Vanilla CFG

There are constituent treebanks for several languages in existence, along with a very limited number of parsing reports on them. The main challenge of constituent parsing for morphologically rich languages is in the handling of the huge number of word forms. According to the reports, the size of the preterminal set in the standard context-free grammar environment is crucial. If we use only the main part-of-speech (POS) tags as preterminals (as is the case with the strongly configurational languages), a considerable amount of information, encoded in the morphological description of the tokens, will be lost. Nevertheless, using the full morphological description as preterminal labels yields a set of over a thousand preterminals, resulting in data sparsity and performance problems [2].

With this in mind, in order to manually construct the Georgian syntactically annotated trees, we had to perform the following text processing procedures: tokenization, morphological analysis, POS tagging and syntactic annotation. Tokenization and morphological analysis were done by the Finite-State Transducer for Georgian [3]. Syntactic annotation procedures were carried out manually using the *Synpathy* tool [4].

The syntactic annotation drew on an adapted version of the TIGER-XML encoding scheme [5]. It takes into account the structural peculiarities of the Georgian language and has been tested in the CLARIN-D project for the GRUG TreeBank repository building [6], [7]. The further issue of the syntactic annotation was based on a version of a Georgian context-free grammar developed at the department of Applied Computational Linguistics, Potsdam University.

In Figure 1, a syntactic tree of a Georgian complex sentence (CS) as an outcome of the Georgian CFG parse procedure is depicted.

თუ ღმერთი გწამთ, არ მითხრათ ახლა, რომ შავი თეთრია.
(Lit. “If you believe in god (=For god’s sake), do not tell me now that black is white”).

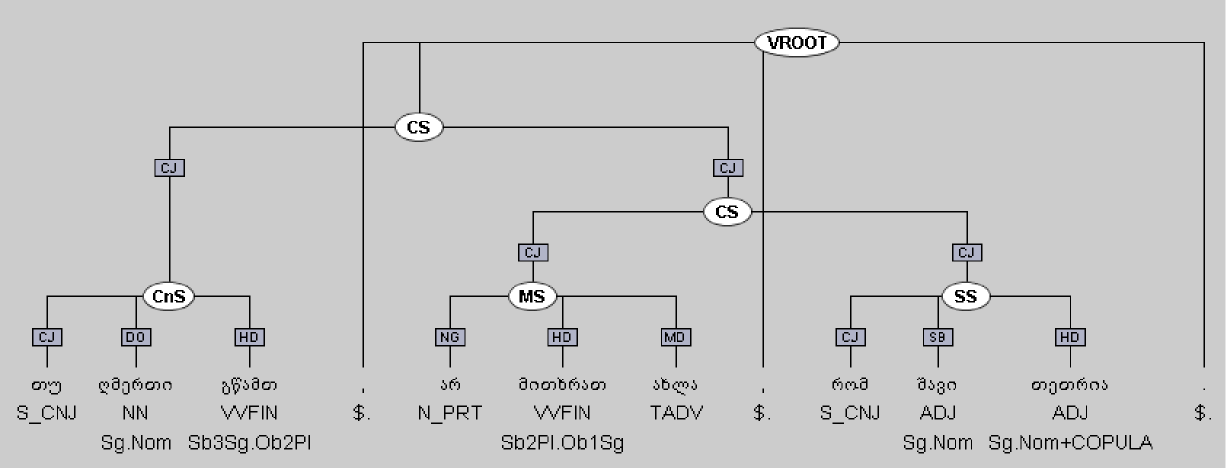

Figure 1. An adapted TIGER-XML scheme for a Georgian sentence.

The sentence in Figure 1 visualizes a hybrid approach to the syntactic annotation procedure as the tree-like graphs and integrates annotation according to the constituency representations and functional relations. In a tree structure the node labels are phrasal categories: Complex Sentence (CS), Main Sentence (MS), Conditional Sentence (CnS), Subordinate Sentence (SS). The edge labels correspond to syntactic functions: Conjunction (CJ), Subject (SB), Head (HD), Modifier (MD), Direct Object (DO).

The tokens in terminal nodes are annotated with POS tags (NN, ADJ, VVFIN, etc.) and supplied with morphological features of case and number for nouns and tense, person and valency for verbs.
We had manually built around 300 high quality morphologically and syntactically annotated trees that have been used as training data for extracting a vanilla CFG and a lexicon for the Georgian language.

4. Future Plans

For building a full-scale Georgian syntactic parser, we intend to make use of the developed vanilla CFG that was extracted from the monolingual Georgian treebank. It will be utilized for finding optimal morphological features/preterminals for implementation in the probabilistic CFG parser. The reason for such a decision is the advantage of a deterministic part-of-speech tagger that can produce a morphologically annotated Georgian corpus achieving almost 100% accuracy after manual disambiguation [2]. Moreover, it has the ability to annotate the tokens with just POS tags, or also with morphological information using features such as case, number, person and tense.

At the first stage, the most successful supervised constituent parsers apply a probabilistic context-free grammar (PCFG) to extract possible parses. The *n*-best list parsers keep just the 50-100 best parses according to the PCFG. These feature templates exploit atomic morphological features and achieve improvements over the standard feature set. These methods use a large feature set — usually a few million features — and are engineered for English [3].

The innovative aspect of the proposed approach is a unique procedure for finding the optimal set of preterminals by merging morphological feature values. The main advantage of this methodology over previous undertakings is the performance speed — it operates inside a PCFG instead of using a parser as a black box with retraining for every evaluation of a feature combination — and it can investigate particular morphological feature values instead of removing a feature with all of its values [3].

References

1. Fraser, A., Schmid, H., Farkas, R., Wang, R. and Schütze, H. Knowledge sources for constituent parsing of German, a morphologically rich and less-configurational language. In *Computational Linguistics*, Volume 39, Issue 1. MIT Press Cambridge, Ma, USA. 2013.
2. Kapanadze, O. Describing Georgian Morphology with a Finite-State System. In *A. Yli-Jura et al. (Eds.): Finite-State Methods and Natural Language Processing 2009, Lecture Notes in Artificial Intelligence*, Volume 6062, pp.114-122, Springer-Verlag, Berlin Heidelberg, 2010.
3. Szántó, Z. and Farkas, R. Special Techniques for Constituent Parsing of Morphologically Rich Languages. *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Gothenburg, Sweden, 2014.
4. *Synpathy: Syntax Editor – Manual* – Nijmegen: Max Planck Institute for Psycholinguistics, 2006.
5. Brants, S. and Hansen, S. Developments in the TIGER Annotation Scheme and their Realization in the Corpus. In *Proceedings of the Third Conference on Language Resources and Evaluation (LREC 2002),* Las Palmas, pp. 1643–1649, 2002.
6. Kapanadze, O. *Multilingual GRUG Parallel TreeBank — Ideas and Methods*. LAMBERT Academic Publisher. 52 p. ISBN-13: 978-3-330-34810-3. EAN: 9783330348103, 2017.
7. *A multilingual German-Russian-Ukrainian-Georgian Parallel Treebank.* http://fedora.clarin-d.uni-saarland.de/grug/