Automatic Morphological Analysis and Syntactic Parsing for the Georgian Language

Oleg Kapanadze
OK'OMPLEX - Innovative Language and Information Technologies, Georgia
ok.omplex@yahoo.com
Nunu Kapanadze
Iv. Javakhishvili Tbilisi State University, Georgia
nunu.kapanadze@tsu.ge
Bogdan Babych
University of Heidelberg, Germany
bogdan.babych@ueid.uni-heidelberg.de
Natia Putkaradze
Iv. Javakhishvili Tbilisi State University, Georgia
natia.putkaradze@tsu.ge

Abstract

In our research, we address Georgian - a language with rich inflectional morphology and with very little fixed structure on the sentence level. The languages of similar design are referred to as Morphologically Rich and Less-Configurational (MR&LC) languages. The present paper concerns issues related to developing crucial NLT tools for the MR&LC Georgian language: a finite state morphological transducer/POS tagger and a Feature-Base Context-Free grammar parser for piping/linking them in an unsupervised shallow syntactic chunker for the Georgian language.

Keywords: Georgian Language, Morphoparser, Feature-Based Grammar, Syntactic Chunking

1 Introduction

This paper presents an undertaking for developing computational applications involving Georgian in order to fill a gap with computationally well-equipped languages and to lower the current scarcity of technological resources for Georgian text processing.

Generalized lexical models do not seem to be as easily adaptable to languages with rich morphology and such a different language-specific property as free word order. We will focus on crucial tools for the Georgian language - a finite state morphological transducer and a Feature-Base Context-Free Grammar (FCFG) parser. Drawing on the syntactic valency property of the verb and language-specific features such as productive morphology, we designed a prototype FCFG parser for automatic syntactic chunking/shallow parsing of the Georgian clause, which we present here.

2 A morphoparser /POS tagger for Georgian

Georgian is an agglutinative language that uses for word form production both suffixing and prefixing. Since morphological analysis is one of the basic concerns for agglutinative languages, a morphoparser is considered as an indispensable computational tool for Georgian text analysis.

The first version of the morphological transducer for Georgian developed a decade ago drew on XEROX finite-state libraries [1]. The basic claim of the finite-state approach is that a morphological analyzer for a natural language can be implemented as a data structure called a Finite-State Transducer [2]. The finite-state approach is built around two practical concepts: constructing lexicographical descriptions of the language using a tool called *Lexc* and expressing morphophonological variations as regular expression rules. *Lexc* supports a simple right-linear morphosyntactic grammar formalism. Consequently, for every morphological description, we have collections of lexicons (lists of morphemes)

that start with a root lexicon. Each morpheme in a lexicon has continuation lexicons, which in turn determine the set of morphemes that can succeed the morpheme [3].

Derived from the Georgian conventional grammar, each collection of lemmas for nouns, pronouns and adjectives is considered a single Root lexicon and the attached numerous suffixes build a correct word form. For example, each entry of the noun Root lexicon can be succeed by a plural morpheme that in turn must be succeeded by a lexicon of seven different case markers possibly followed by postfixes. Some of those morphemes can be followed by so-called emphatic vocal and emphatic particles at the end of a word form. In contrast to the sketched scheme, for building a finite verb form suffixes as well as prefixes are needed. The mentioned XEROX finite-state Georgian morphoparser draws on a scheme in which each verb Root lexicon entry is proceeded by three rows of prefixes and followed by possible five different sets of suffixes. However, in exceptional cases, a finite verb form may appear also as a verbal Root lexicon entry without inflexion, when affixes are represented by zero allomorphs from three prefixes and consequent five suffixes' ranks.

After reimplementation, to date there are three versions of the Georgian FST morphoparser with different output formats: First - with TIGER-XML format [4] as input for the Georgian syntactically annotated treebank building. Second - as input for different NLP tools with lemmas of parsed words in the input text. And the third - a simplified option without the lemmatized input words. Each of the listed versions is capable to tokenize, POS-tag, and morphologically annotate the input Georgian text. After a light manual post-editing the morphoparser output can reach up to almost 100% accuracy.

3 A syntactic chunker for unsupervised parsing of the Georgian clause

Shallow syntactic parsing (also known as Chunking) aims at identifying syntactic constituents like a noun or a verb phrase within a clause. Theoretically, it can consist of a VP (Verb Phrase) and n mandatory NPs. The number of NPs in the clause (resp. graph/tree) is determined by the syntactic valency (VAL) parameter of the Georgian Finite Verb Form. VAL is inferred in the process of morphological analysis of the Georgian text since a “static” source in the form of a lexicon with VAL data for a specific verb stem would not work. The reason is that the syntactic valency frame may change according to the series as suggested in Table 1. Therefore, for each finite verb, it is determined “on the fly” in the process of morphological analysis.

After clustering the Georgian verb stock according to the valency data/syntactic frames, one can perform automatic shallow parsing to produce chunks of text as NPs. They can be constructed as the maximal projection (using the longest match rule) of a head word_ j (normally noun or equivalent) of the corresponding NP_j.

Pursuant to syntactic valency and case marker distributions of NPs across the three series adopted in the Georgian academic grammar, a table is suggested that specifies syntactic frames for ten clusters of Transitive (TV) and Intransitive (ITV) verb sets:

Cluster	I serie	II serie	III serie
1 / ITV	VAL=N	VAL=E	VAL=D
2 / ITV	VAL=ND	VAL=ED	VAL=D(D+postf)
3 / TV	VAL=ND	VAL=ED	VAL=DN
4 / TV	VAL=NDD	VAL=EDD	VAL=DN (D+postf)
5 / ITV	VAL=N	VAL=N	VAL=N
6 / ITV	VAL=ND	VAL=ND	VAL=ND
7 / ITV	VAL=ND	VAL=ND	VAL=DN
8/ ITV	VAL=DN	VAL=DN	VAL=DN
9 / ITV	VAL=D	VAL=D	VAL=D
10 / ITV	VAL=0	VAL=0	VAL=0

Table 1. Syntactic frame distribution for Georgian verb sets.

In the table, the valency parameter (VAL) points to verb arguments (resp. syntactic valency). Their number can vary depending on the verb transitivity and series. As it can be observed, VAL for cluster 2/ITV in the I serie is bivalent with head word in N(ominative) for Subject and in D(ative) for Indirect Object. Consequently, VAL is N(ominative) and E(rgative) in the II serie. In the III serie it is reduced to

a monovalent verb as the second argument in D(ative) is inflected with a postfix (D+postf) and it may not be identified as a syntactic place holder or syntactic argument in the verbal frame.

The same phenomenon can be traced also for cluster 4/TV with transitive verbs, which from a trivalent option in the I and II series is represented as a bivalent invariant in the III serie with (D+postf).

As sketched above, the capital letters (N,E,D) in VAL parameter stand for the case markers of the head words in consequent NP phrases, which number (n) in a clause can vary as [0,1,2,3]:

If n=1, the consequent NP phrase as a maximal projection of its head word is assigned a syntactic function of *Subject*.

If *n =2* with a transitive verb, clausal constituents NP1 and NP2 (derived from consequent head words' case markers) will be labelled as maximal projections of a Subject and a *Direct_Object*.

For *n=2* with an intransitive verb, NP1 and NP2 will be distributed as maximal projections of a *Subject* and a *Direct_Object*.

With *n =3*, by default, NP1, NP2 and NP3, depending on their consequent head word's case formants, are the longest matches of a *Subject*, a *Direct_Object* and an *Indirect_Object* phrasal constituents.

Clauses with zero (*'0'*) syntactic valency denoting the natural phenomena are without any phrasal constituent (Cluster 14 in the table). We already mentioned that in some VAL parameter samples the parentheses contain the case marker with postfix (D+postf) as the head word for NPs that may appear for specific verb clusters (1,4,5) in the III serie presented in the Syntactic frame distribution in Table 1.

The verb Transitivity in the suggested shallow parsing scheme is a redundant feature. It is included in Table 1 just to refer to the grammatical characteristics used in traditional syntactic analysis systems.

Once all the mandatory and optional constituents as NP's maximal projections are identified in a clause/sentence, drawing on Lexical Production Rules (LPR) a syntactic parser can produce a shallow syntactic parse tree of a clause/sentence under analysis.

Based on the sketched principles and compiled Grammar with the LPR of the Georgian language, the NLTK libraries come into play for the automatic construction of morphologically and syntactically annotated shallow parse trees. LPR rules can be created completely manually or semi-automatically. The last option anticipates that a parser automatically assigns some syntactic structure to a text which eventually is checked by linguists and, if necessary, corrected.

The number of NPs in a clause (resp. graph/tree) is determined by the syntactic valency of a head verb of the clause, identified by the morphoparser and percolated as a feature in VP. The annotated text is usually in a form of syntactic bracket labelling so that the pre-processed sentence will be grouped into a hierarchical form of phrase structure.

A practical feasibility of the sketched scheme was checked out with an open-source library called Natural Language Toolkit (NLTK) based on the Python programming language [5]. In Fig. 1 is depicted an NLTK output of Feature-Based Context-Free parser for a Georgian clause:

ყველა მამაცი მეომარი ძალიან თავგანწირვით იცავს ყველა სუსტ ადამიანს.
(Lit. "Every brave warrior with great sacrifice protects all weak people")

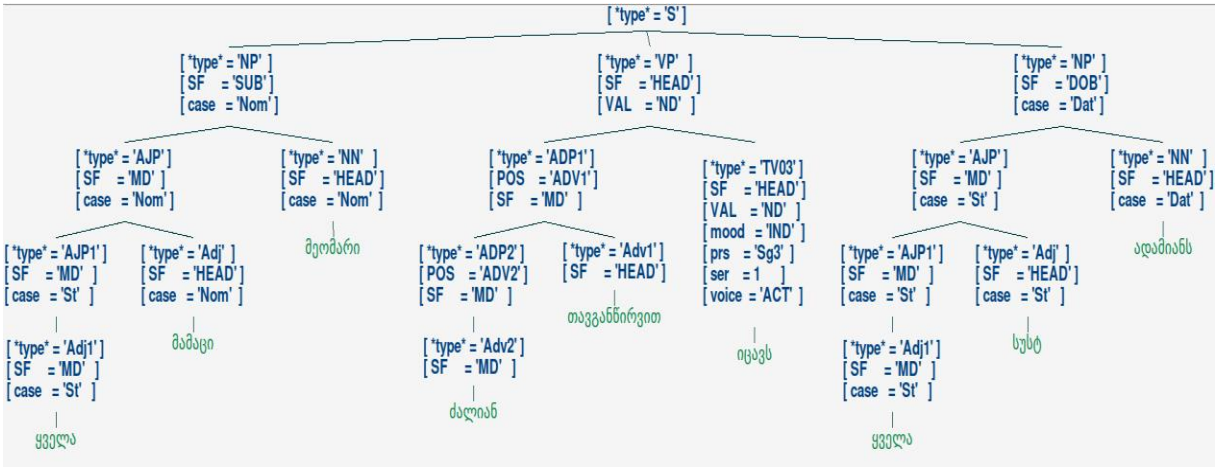

Fig.1. NLTK parser output of Feature-Based Context-Free parser for a Georgian clause

The NLTK output consists of powerful database-oriented representations for graph structures in which each leaf (= token) and each node (= linguistic constituent) has a unique identifier. For the Georgian clause, it visualizes a hybrid approach to the syntactic annotation issue as tree-like graphs and integrates annotation according to the constituency representations and functional relations. The graph nodes reflect two sorts of rules instantiated in the Feature-Based Context-Free Grammar: the phrasal-level grammar rules and lexical rules. The last one builds nodes saturated with a bunch of different features and morpho-syntactic class labels (resp. POS-tags). The 'type' feature on a clausal level denotes phrasal labels (S, VP. NP). The same nodes are supplied also with SF (Syntactic Function) indicators such as SUB (Subject), DO (Direct Object), and HEAD (Head of clause/phrase).

The intermediate-level nodes with SF are MD (Modifiers). VP node contains a grammatical feature TENSE with a value 'pres' (Present) and val='ND' that are percolated from the terminal node of 'type'='TV03' (Cluster 3 Transitive Verb). A value 'ND' of val-feature (syntactic valency) points to a case marker (case='*N*[ominative]') of the head word in the left NP with syntactic function (SF='SUB') and to a case marker (case='*D*[ative]') of the head word in the right NP as SF='DO' (Direct_Object).

The general scheme of the graph for phrasal categories and their syntactic relations in the clause is built in the X-Bar projection tradition [6]. The clause's predicate-argument structure (syntactic functions) is based on the syntactic valency grammatical concept, an analogue of the "head feature principle" in Head-Driven Phrase Structure Grammar [7].

4 Conclusion and the Future Plans

In the presented paper, we have featured an option for developing a syntactic chunker/shallow parser for the Georgian language.

As the initial step to the syntactic analysis, we reimplemented the rule-based Finite-State Morphological Transducer for Georgian text morphological analysis, lemmatization and POS tagging. To build an interface between the TIGER XML scheme and an input format for NLTK, we had to disambiguate manually and reformat the output of the Georgian morphoparser.

As a necessary step in the syntactic valency-driven Feature-Based Grammar parser implementation, we have studied the Georgian verb stock and clustered it according to syntactic valency features. Ten verb clusters with different valency distributions, bound with syntactic frames, are identified to the date. For each cluster, we developed and started training a prototype Feature-Based Grammar version for Georgian.

As a syntactic parsing testbed, we have utilized a broadly recognized open-source library NLTK developed using the Python programming language.

In the meantime, we are developing a converter module capable of porting automatically the output of the morphoparser at hand into the acceptable format for the NLTK input engine. This would provide an option for piping the morphological transducer with the Feature-Based syntactic parser for unsupervised syntactic chunking of the Georgian text.

Reference

1. Kapanadze, O. (2010). 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.
2. Beesley, K. R. and Karttunen L. (2003). Finite State Morphology. CSLI Publications. 2003.
3. Drobac, S., Lindén, K., Pirinen, T.A. and Silfverberg, M. (2014). Heuristic Hyper-minimization of Finite State Lexicons. LREC, 2014. La Valetta, Malta.
4. Brants, S. and Hansen, S. (2000). 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.
5. Bird, S., Loper, E. and Klein, E. (2009). Natural Language Processing with Python. O'Reilly Media Inc. 2009.
6. Kornai, A. (1990). The X-bar theory of phrase structure, Language, 66(1), pp. 24–50, 1990.
7. Pollard C. and Sag I.A. (1994). Head-Driven Phrase- Structure Grammar. Chicago. University of Chicago Press. 1994.