The creation of large-scale annotated corpora of minority languages using UniParser and the EANC platform

Timofey ARKHANGELSKIY[1] *Oleg BELYAEV*[2,3] *Arseniy VYDRIN*[4]
(1) National Research University Higher Scool of Economics,
Moscow, Myasnitskaya St. 20
(2) Institute of Linguistics of the Russian Academy of Sciences,
Moscow, Bolshoy Kislovskiy Lane 1/1
(3) Sholokhov Moscow State University for the Humanities,
Moscow, 3-ya Vladimirskaya St. 5, room 35
(4) Institute for Linguistic Research of the Russian Academy of Sciences,
St. Petersburg, Tuchkov per. 9
tarkhangelskiy@hse.ru, belyaev@iling-ran.ru, senjacom@gmail.com

Abstract

This paper is devoted to the use of two tools for creating morphologically annotated linguistic corpora: UniParser and the EANC platform. The EANC platform is the database and search framework originally developed for the Eastern Armenian National Corpus (www.eanc.net) and later adopted for other languages. UniParser is an automated morphological analysis tool developed specifically for creating corpora of languages with relatively small numbers of native speakers for which the development of parsers from scratch is not feasible. It has been designed for use with the EANC platform and generates XML output in the EANC format.

UniParser and the EANC platform have already been used for the creation of the corpora of several languages: Albanian, Kalmyk, Lezgian, Ossetic, of which the Ossetic corpus is the largest (5 million tokens, 10 million planned for 2013), and are currently being employed in construction of the corpora of Buryat and Modern Greek languages. This paper will describe the general architecture of the EANC platform and UniParser, providing the Ossetic corpus as an example of the advantages and disadvantages of the described approach.

Keywords : corpus linguistics, automated morphological analysis, language documentation, Iranian languages, Ossetic

1 Corpus technologies and minority languages

Corpus lingustics is currently a rapidly developing area of study. Corpora created for such large languages as English, Czech, or Russian are being increasingly used for analyzing the grammatical phenomena of these languages drawing on more empirical material than could ever be possible before in the history of linguistics. Using corpus data as a basis for linguistic research has become a new "philosophical approach" rather than just one of possible methodologies (Leech 1991) and is widely considered to be superior to the classical approaches of introspection/elicitation, since it draws on real language use instead of artificially constructed examples.

Unfortunately, the creation of a reasonably large annotated corpus (with 1 million tokens or more), especially for a morphologically rich language, is a complicated task that few languages can "afford". The prerequisites for creating a successful corpus are: (1) the availability of digitized texts in that language; (2) the existence of an automatic morphological analyzer. Both of these tasks require considerable investment of time and money, even when a language has a reasonably developed literary tradition (like e.g. Ossetic does, having had literature since the late 1800s).

Therefore, the linguistic community ends up in a situation when large corpora suitable for efficiently studying grammatical phenomena are available only for the major languages of the world. This creates a strong typological bias in favour of these languages.

Our work on Ossetic is an attempt to overcome this limitation, producing a large corpus of a minority language of Russia. Ossetic is an Iranian (Indo-European) language spoken by about 500,000 people mainly in the Russian Federation, in the Republic of North Ossetia situated in the North Caucasus. Digitized versions of Ossetic literature (written in the literary Iron dialect) are readily available from publishers in Vladikavkaz[1]. However, problematic was the creating of an automatic morphological analyzer for Ossetic, a language with a relatively rich inflectional morphology (9 nominal cases and a large number of verbal forms), and the choice of a web platform to be used for accessing the corpus. The solution was reached by developing a universal morphological parser. It operates using rules provided by linguists (and can thus be applied to different languages) and produces XML output that is accepted by the Eastern Armenian National Corpus (EANC) platform, which was adapted for use with the Ossetic language. The final result is the Ossetic National Corpus, which is freely available online (http://corpus.ossetic-

[1] We are thankful to the editors of the *Max dug* literary journal, as well as to the personnel of the *Ir* publishing house, for providing us with electronic versions of their publications and approving of them being used in the corpus.

studies.org) and contains about 5 million morphologically analyzed tokens (with 10 million planned for 2013).

2 The EANC platform

The platform we used for the Ossetic corpus was initially developed by CorpusTechnologies for the Eastern Armenian National Corpus in 2007 (see e. g. Khurshudian et al. 2009). It includes a search engine and a web interface. Although the interface was designed specifically for the Armenian language, the search engine itself is language-independent and is suitable for a great variety of languages. To use the platform with Ossetic, we had to produce parsed texts in the format supported by the EANC platform and make some corrections to the user interface.

2.1 General architecture and features

Parsed data are stored in a number of datafiles. The text itself is stored in XML form in these datafiles. There is a number of index files which list positions of all occurrences of specific wordforms, lemmas and combinations of grammatical tags in the texts. The texts can be equipped with metadata, such as the name of the text, the name of the author, the date of creation, etc. The datafiles are produced by the indexer written in Python which takes parsed texts as its input.

The user interface is written in PHP and HTML. When the user initiates a new search, the interface collects the data entered by the user and sends them to the client written in PHP which, in turn, transmits the query to the server. The server is a program written in C++ which is constantly running and waiting for requests. The server performs the search and sends the result back to the client. Then the result is transformed according to the specified display options and displayed in the browser.

The main query types offered by the EANC interface are wordform, lemma, and grammatical tags queries. When searching for a particular wordform, lemma, or a set of grammatical tags, the platform displays all sentences containing the requested wordforms.

There are a number of special characters for enhanced queries. Specifically, one can use "*" for arbitrary number of characters, "|" for disjunction, "&" for conjunction, and "~" for negation. For example, if "*тты|*тыл" is typed into the Wordform box in the Ossetic corpus, the platform will find all wordforms ending with either *тты* or *тыл*. In the case more than one operator is used, their order can be specified by means of parentheses.

There are also other restrictions one may impose on the words one wants to find. They include specifying a subcorpus, restrictions on the positions of the words being searched relative to each other, etc. The output can be displayed in several ways, including KWIC (Key Word In Context), and supports transliteration mode.

2.2 Adopting the platform for other languages

To use the EANC platform with another corpus, we had to rewrite the language-specific parts of the user interface. These include, for example, a form where one can specify grammar query with the help of checkboxes, every checkbox corresponding to one of the grammatical tags used in the corpus, such as "inessive case". To do that in a more efficient way, we wrote a Python script which takes a simple csv table with all grammatical features of the language and transforms it into the PHP file used in the interface. Parts concerning the writing system were also rewritten, namely the transliteration system and the virtual keyboard. The indexer and the datafile system remained intact and works fine with languages other than Armenian without additional adjustments, provided one doesn't need any additional features.

3 UniParser

A corpus of texts without annotation is effectively a mere electronic library with quite limited applicability for linguistic research. Annotation can include different kinds of data – among others, it can include text-level information such as the name of the author and the time of creation of the text, sentence-level, or word-level information.

The most widely used type of word-by-word annotation used in general purpose corpora, such as the one under consideration, is lemmatization and grammatical markup. Lemmatization means that for every wordform in the corpus, its lemma (dictionary form) is provided, and grammatical markup of a wordform means that all or some subset of grammatical values expressed in it are explicitly shown. (The "lighter" variant of the latter is part-of-speech tagging.) While for languages with poor morphology, such as English, the absence of grammatical markup might not constitute a big obstacle, for morphologically rich languages such as Ossetic it would make any corpus research involving searches for all instances of a given lexeme or all wordforms with a given morphosyntactic feature virtually impossible.

While for small corpora whose size doesn't exceed several hundred thousand tokens it is feasible to annotate them manually, with corpus size going in the millions, automatic tagging is the only possible way of performing this task. Ossetic corpus being approximately 5 million tokens in size at the beginning with even more on the way, we needed an automated utility to annotate with. In the same time, we also were in contact with several other related groups working on other corpora of morphologically rich and diverse languages, namely Albanian, Greek, Kalmyk, and Lezgian, who were also in need of such a tool, so rather than creating a program designed specifically for morphological parsing of Ossetic language, it was deemed more feasible to build a general parser with capability of parsing any of those languages provided their lexicons and grammars are described in some suitable format.

After considering several existing resources, we decided to develop a new system from scratch. The resulting tool, UniParser, is suitable for parsing large amounts of texts in structurally different languages. We are going to make UniParser freely available in 2013. The tool and the format it utilizes to store the information about the language, are the subject of the description below.

3.1 The requirements

When designing a parsing tool for middle-sized and large corpora in different languages, we had in mind several requirements it should conform to.

First, it should work fast enough to cope with big amounts of texts. This is one of the reasons why we couldn't use tools aimed at parsing small corpora. For example, one of the parsers used in the Fieldworks Language Explorer, XAMPLE, and its predecessor, AMPLE, can process tens to hundreds wordforms per second (see Black, Simons 2006), which would require at least half a day for parsing the entire Ossetic Corpus.

Second, the files should have simple enough format to be edited with the help of an ordinary plain text processor by a linguist without programming or other technical skills. The only specific piece of knowledge which might be required for describing some fragments of grammar in UniParser format is regular expressions.

Third, it should be flexible enough to be used with structurally different languages, addressing a wide range of various morphological phenomena.

Other requirements, namely limitation imposed on the morpological model and the output format of the parsed text are presented in more detail below.

3.2 The morphological model

The basic approach taken in the UniParser format can be roughly described as Word-And-Paradigm morphology (see e. g. (Matthews, 1972) for thorough description of this model). Here by this term we mean that wordforms in the parsed corpus should be labelled with grammatical tags like "Noun" or "genitive case", and provided with lemmata, but the researchers who compile the corpus shouldn't be obliged to overtly mark morpheme boundaries when making a description of the grammar. This is contrary to the approach taken e. g. in the parsers used in the Fieldworks Language Explorer where the user first has to create a dictionary of morphemes and then define templates describing the ways these morphemes can be assembled together producing wordforms. However, if the user wants their corpus to be glossed and displayed with interlinears, the UniParser format offers a possibility of doing that. We adhere to this approach to facilitate the annotation of corpora in flective languages where division into morphemes may be not that straightforward, so that providing accurate information about individual morphemes in the grammatical description would be time-consuming and often subjective.

The word inflection in the UniParser format is described, first of all, through the notions of stem, inflexion, paradigm, and productive derivation model. All data in the description of the lexicon and the grammar is concerned only with the graphical representations of wordforms, without appealing to their phonemic or any other "deep" levels.

A lexeme is thought of as a set of wordforms. In the vast number of cases, all these forms have some letters in common. Thus every wordform of a lexeme can be divided into the part common for the entire lexeme and the part that is unique for the given form (or several forms). These units are called a stem and an inflexion. If a unit is disjoint, the places where parts of another units can appear are marked with dots. The dot also appears at the beginning or at the end of a unit if it can be, respectively, preceded or followed by a part of another unit. So, in Ossetic (and in most other IE languages) most stems would have a dot at the end meaning that they can take inflectional markers on the right. Accordingly, most inflexions would look like a contiguous block of letters with a dot at the beginning. A stem and an inflexion can be combined into a wordform by inserting parts of one of them into the dot-marked slots of the other. To take an extreme example, in Arabic wordform *katabtu* 'I wrote' with the stem KTB, the stem would be written as ".k.t.b.", and the inflexion as ".a.a.tu".

A complete set of inflexions a lexeme can take is called a paradigm. Different lexemes of the same part of speech can belong to different paradigms and use different markers for expressing the same grammatical values. Every inflexion in the UniParser format belongs to one of the paradigms.

Another feature of UniParser format is productive derivation models. By derivation we understand the process whereby new lexemes are created on the basis of existing ones according to some rules; a productive derivation model is such a rule which is applicable to a large and open set of lexemes (say, to all lexemes of a particular paradigm type). For example, many Ossetic verbs have perfectivized forms with different preverbs which are considered separate lexemes. A productive derivation model was set up for every preverb, which automatically adds all the derivatives to the lexicon.

3.3 Dictionary format

All the information about the lexicon and the grammar of a language is stored in a number of files, the core files being stem.txt (lexicon), paradigm.txt (inflexions) and derivation.txt (productive derivation models).

The format of description is based on YAML, which was preferred over XML because the former is more human-readable, so that the files can be edited by hand. All files contain "objects" which are collections of parameter-value pairs, values being strings or another objects.

The basic object of the file stem.txt is a lexeme, which is described as a list of parameter-value pairs. This list is open in principle, with only several fields being obligatory, namely lex (the lemma), stem, paradigm, and gramm (grammatical tags which should be assigned to every wordform of that lexeme in the parsed text). In the Ossetic dictionary, the two additional fields contain Russian and English translations of the lemma.

In the case of suppletivism or stem alternations, several stems (allomorphs of the stem) can be stipulated instead of one. Another case when several stems can be stipulated is free variation. As an example of a lexeme with both of these phenomena, we will take the Ossetic word æххормаг 'hunger' which has three stem allomorphs, each allomorph possessing two variants:

```
-lexeme
 lex: æххормаг
 stem: æххормаг.//ххормаг.|æххормадж.//ххормадж.|
æххормæг.//ххормæг.
 paradigm: Nct
 gramm: N-ADJ,inanim,nonhuman
```

The basic object of the file paradigm.txt is a paradigm which is a collection of inflexions. A fragment of Ossetic nominal paradigm Nctt is presented below:

```
-paradigm: Nctt
 -flex: <1>.ы
  gramm: sg,gen
  gloss: GEN
 -flex: <0>.æн
  gramm: sg,dat
  gloss: DAT
```

The number in angle brackets defines the stem allomorph a given inflexion can be used with. In inflexions, the only obligatory field is gramm which contains grammatical tags assigned to all wordforms with that inflexion by the parser. If the user wants the text to be glossed, she may optionally specify the division of the inflexion into morphemes and add the gloss field.

3.4 Technical details

The UniParser tool consists of a simple user interface, the preprocessing module and the analysis module. The user interface allows to load description files, view full paradigms of the lexemes in the lexicon (which is crucial for error-checking), and launch preprocessing or analysis. The preprocessing module transforms the description of the language into a datafile to be used in the course of parsing. The analysis module uses a finite-state automaton with hashtables. The analysis module of the UniParser tool was implemented in C++, and the user interface and the preprocessing module were written in C#.

The parsing speed for Ossetic texts reached approximately 7000 wordforms per second on an AMD Athlon 64X2 (2x2,20 GHz) system with 2 GB RAM. By using a relatively short list of pre-analyzed high-frequency wordforms, we could increase the speed some 30% further. Although the speed can be considered sufficiently high for our purposes (12 minutes for the whole corpus), there is evidently room for improvement (for example, by introducing multithreading). Another parameter which should be optimised is memory usage, as in the current version more than 1 GB of memory was used.

No statistical disambiguation techniques were used because, despite their high accuracy rates, there is a risk of systematically distorting some linguistically peculiar information. Therefore any token was assigned all parses that were possible on the basis of the language description. The quality of analysis can be estimated by parsed tokens rate and the average number of parses per parsed token. Among all the tokens of the corpus, more than 85% were assigned at least one parse, the dictionary size being about 15,000 entries. The average number of parses per parsed token is approximately 1.7. The figure is quite high, so addressing this problem with the help of deterministic disambiguation rules is planned.

The parser takes plain text files encoded in UTF-8 as its input and produces an XML file with the parsed text. The XML we use is similar to that used in the Russian National Corpus.

Conclusion and perspectives

As a result of developing a universal morphological parser and a set of rules for this parser, as well as adopting an existing search engine (the EANC platform) for being used with the Ossetic language, we have successfully created a corpus of literary Ossetic consisting of 5 million tokens, which is one of the first corpora of such scale having been developed for a minority language. Our next aim is to reach 10 million tokens, as well as develop the parser further in order to allow for analyzing compounds and verbs with incorporated nouns, which are quite widespread in Ossetic. This will allow us to reach higher percentages of analyzed tokens than the current 85%. A further possible area of inquiry is developing mechanisms for automatic resolution of ambiguity, at least in those cases where the function of the wordform is clear from its immediate context.

References

Black, H. A. and Simons, G. F. (2006). The SIL FieldWorks Language Explorer Approach to Morphological Parsing. In *Computational Linguistics for Less-studied Languages: Proceedings of Texas Linguistics Society 10, 3–5 November 2006*, Austin, TX

Khurshudian, V. G., Daniel, M. A., Levonian D. V., Plungian V. A., Polyakov A. E., Rubakov S. V. (2009). Eastern Armenian National Corpus. In *Computational Linguistics and Intellectual Technologies (Papers from the Annual International Conference "Dialogue 2009"),* 8 (15), pages 509–518, Moscow, RGGU

Leech, G. (1992). Corpora and theories of linguistic performance. In *J. Svartvik (ed)., Directions in Corpus Linguistics: Proceedings of the Nobel Symposium 82, Stockholm, 4–8 August 1991*, pages 105–122, Berlin, Mouton de Gruyter

Matthews, P. H. (1972). *Inflectional Morphology (a Theoretical Study based on Aspects of Latin Verb Conjugation).* Cambridge, Cambridge University Press