Synchronized Mediawiki based analyzer dictionary development

Jack Rueter
University of Helsinki
Department of Modern Languages
jack.rueter@helsinki.fi

Mika Hämäläinen
University of Helsinki
Department of Computer Science
mika.hamalainen@helsinki.fi

Abstract

Open-source analyzer dictionary development is being implemented for Skolt Sami, Ingrian, Moksha-Mordvin, etc. in the Helsinki CSC infrastructure; home of the Finnish Kielipankki 'Language Bank' and Termipankki 'Term Bank'. The proximity of minority-language corpora in need of annotation and the multiple usage of controlled wikimedia-type dictionaries make CSC an attractive site for synchronized transducer dictionary development. The open-source FST development of Uralic and other minority languages at Giellatekno-Divvun in Tromsø demonstrates a vast potential for reusage of FST-s, only augmented by open-source work in OmorFi, Apertium and Universal Dependency <http://universaldependencies.org/#language-urj>. The initial idea is to allow synchronized editing of Giellatekno XML and CSC Wiki structures via github. In addition to allowing for simple lexc LEMMA:STEM CONTINUATION_LEXICON "TRANSLATION" ; line exports, the parallel dictionaries will provide for documentation of derivation, morpho-syntactic information on valency and government, semantics and etymology.

The 3rd International Workshop for Computational Linguistics of Uralic Languages, pages 1–7,
St. Petersburg, Russia, 23–24 January 2017.

1 Introduction

Open-source finite-state transducer development and application as we know it today in the Giellatekno infrastructure[1] at Tromsø, Norway dates back to the early 1990s. It begins with the morphological description of different Sami languages, grammatical analysis and syntax. Morphological and morphosyntactic description lays the foundation for tool building, such as Divvun[2], and working solutions attract soft coding for the application of research and tools to other languages. Open-source compilers from HFST[3] in Helsinki, are gradually worked into the infrastructure after 2008. With the growth of the research community and tool building diversity comes the practicality of reusable resources, descriptions and testing formalisms. As things improve the number of uses and users also increases. One resource, in particular, is the four-fold combination of lemma, stem, pos/continuation lexicon and gloss, afforded by many of the language projects at Giellatekno in lexc code.

Lexc code containing multiple nodes of information can be stored in XML files for xsl transformation and project-specific transducer construction. The need for multiple transducers presents itself when tagging strategies are not shared by the multiple projects of a given language. Although the two-level model may be sufficient for all projects, normative labeling, morphosyntactic information and semantic tag needs will vary for TTS, MT, ICALL, spellcheckers and other morphological analyzer projects. The Giellatekno-Divvun strategy has been to utilize one lexc and twolc in all North Sami projects with filters for selecting the necessary code. For this multiple and iterated use of resources, on the contrary, the solution presents itself in XML-format analyzer dictionaries with multiple xsl transformations and the possibility to keep up with semantic wiki strategies being adopted in open wikimedia projects[4].

At CSC in Helsinki, Finland, a sanat-server has been put into operation and provides access, initially, to the Kielipankki[5] (Finnish Language Bank) wordnet and Ludic dictionary development. The Ludic dictionary provides for Finnish-Ludic and Russian-Ludic documentation of the Ludic language in a Wikimedia environment. The Wikimedia environment is a sibling of what is used for facilitating Termipankki[6] (the Finnish Term Bank); domains and subdomains can be established for administrating access and editing rights. This type of environment is desirable for synchronic editing strategies involving XML and php input. Unlike wiktionary and wikipedia

[1] http://giellatekno.uit.no/

[2] http://divvun.no/

[3] http://www.ling.helsinki.fi/kieliteknologia/tutkimus/hfst/
Lene Antonsen (p.c. 2016)

[4] https://en.wiktionary.org/wiki/Wiktionary:Semantic_relations/

[5] https://www.kielipankki.fi/

[6] http://tieteentermipankki.fi/

instances, however, a wikimedia environment at CSC might be afforded better control of vital inflection-type data and allow for controlled levels of editing and citizen science.

2 XML Dictionaries

XML analyzer dictionaries initially consist of the three bare necessities for the lexc files. They contain lemma, stem and continuation class information. Additional semantic tags, as well as usage and POS information is also stored in the xml. The structure of the Skolt Sami XML dictionary at Giellatekno, in fact, goes much further but has yet to reach a level of consistency, which is due to multiple contributions from various projects. The Skolt Sami dictionary contains many fields of information and references. There are external references to audiofiles, Swadesh list instances[7] and etymological databases[8], while at the same time there are internal references to derivation, compounding, government and semantics; both ICALL and multilingual translation resources.

By integrating morphology, morphosyntax and semantics into one system, a direct link can be drawn to research dictionaries and syntactic contexts available in published fieldwork research. Utilization of research results leads of modeling XML structure after the derivation-oriented nature of research dictionaries. It directs development toward semi-automated derivation and compounding documentation. Research-language definitions (Russian, German and Finnish) provide new dimensions for tandem development of multilingual resources wherein the semantics provided by wordnet mapping[9] of majority languages can be utilized in the prediction and implementation of semantic alignment in minority languages. Synchronized tandem development presupposes multi-input strategies for different projects at the single or multiple-language level, as well as an opportunity for citizen-science contribution.

At present the XML element structure contains a two-way division: one lemmaGroup <lg/> and one or more morphosyntacticGroups <mg/>. The <lg/> is associated with lemma, stem, inflection type, audio files, etymology links and derivational

Tommi Pirinen, p.c. 2015

[7] http://dla.library.upenn.edu/dla/olac/search.html?fq=subject_language_facet%3A”Skolt Sami”/

[8] http://kaino.kotus.fi/algu/index.php?t=etusivu&kkieli=en

In the spring of 2016, the Finno-Ugrian Society decided to make many of its early publications in fieldwork and research searchable, among other things this meant the OCR of 8 volumes of Mordvin Folklore, consisting of transcriptions with parallel German and possibly Russian translations, as well as a 2703-page dialect dictionary for the two literary languages.

[9] https://www.academia.edu/13230592/Developing_electronic_lexical_resources_for_Saami_Languages/

or compounding data. The <mg/>, however, is associated with semantics, dependencies and argument structure, as well as translations and contextual examples. There is also room for source information in many of the elements as will be noted in the sanat.csc.fi/wiki/Luokka:Sms/ entries.

3 The XML-Wiki Synchronization System and its Architecture

Because dictionary development can be done on both XML and Wiki side simultaneously, a database system is needed in between through which the latest changes on both sides can be synchronized so that the people editing the XML files have the same version as the people modifying the Mediawiki dictionary. Furthermore, data can come to the dictionary from different XML files, all of them having a slightly different structure. For instance, in the case of Skolt Sami, there are three different kinds of XML files that all need to be synchronized with the system.

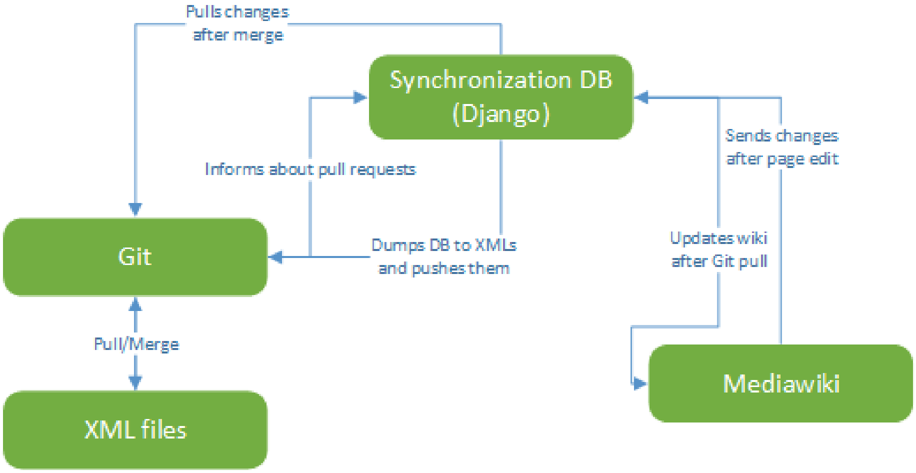
Pulls changes after merge
Synchronization DB (Django)
Informs about pull requests
Sends changes after page edit
Git
Dumps DB to XMLs and pushes them
Updates wiki after Git pull
Pull/Merge
Mediawiki
XML files

Figure 1: System architecture

The architecture of the system is described in the Figure 1. On one end, there are the XML files and the other Mediawiki. In the middle, there's a Django based service with its own Mongo based database through which the synchronization is done. Mediawiki is directly linked to the Django service, so that all the changes will be reflected to both systems directly.

A problem arising from this architecture is how conflicts are resolved. That is why only the Mediawiki side is directly linked to the Django service. The XML files need to be updated through Git. The work flow for the people working with the XML files is the following: they make modifications to the XML files in their own branches and once they are ready to release the changes to the Mediawiki side, they will create a pull request for that branch. What it means is that they want to merge their edits to the master branch which contains the production data. At this point, an automatic Travis script is executed and what it essentially does, is that it makes the Django service dump its database into XML format to the Git project, commit the possible changes and push them to the master branch. If the dictionary has been modified in the Mediawiki side, the person developing the XML files has to resolve the conflicts using Git. Because resolving conflicts is what Git is really good at, leaving the conflict resolution to that system makes our system easier to develop and maintain as we can safely assume that Git is stable enough. Once the pull request is finished and all the conflicts are resolved, the Django based system will be notified about the changes in the master branch and it will start to update its own database and Mediawiki.

Another challenge for this system is the different data structures used. XMLs have their own formatting, the Django services internal database uses Mongo which is essentially a JSON based data structure and finally Mediawiki shows its pages using its own Wiki syntax. The Django services data structure cannot be identical with the XMLs for two different reasons: first of all, even Skolt Sami has multiple different XMLs with different structures and secondly we aim to support multiple languages and make the process of integrating new languages an easy one. As a result of our architecture choice, what is mostly needed for a new language to be added to the system is an XML to/from JSON conversion script. This way we can reuse the existing Mediawiki to/from JSON conversion code, so that the Mediawiki and Django service sides can communicate with each other already when a new language with XML files is added. Thus far, this has seemed to work well when the Ingrian XML files were added to the system.

The system also needs to be easy to use for non-technical people. From people working directly with the XML files, you can expect that they know how to maintain the structure of the files, for example that every open tag has to be closed in the correct order and so on. However, from people dealing with the Mediawiki based dictionary, you cannot expect them to understand how the Wiki page needs to be structured in order for the Django service to understand it correctly. That is why when a user enters the page edit mode on Mediawiki, a form is displayed instead of the raw Wiki text. This form then turns the edits into Wiki text with Java Script so that the Mediawiki platform can display the dictionary entry correctly. The same Java Script is used by the Django service when it updates the pages on Mediawiki.

4 Wiki Dictionary

The Wiki-based dictionary is hosted on CSC's servers (https://sanat.csc.fi). Before this project their Mediawiki platform already had one dictionary, Ludic, and the Finnish WordNet. In addition to that, the Institute for the Languages in Finland (http://www.kotus.fi/en) has expressed their interest in adding their own dictionaries and word lists to the same Mediawiki platform. Since multiple parties are interested in the same platform, it only makes sense for us to use it as well. This opens the possibility of interlinking dictionaries from different providers to enhance the quality of all of them.

From the point of view of our project, the Mediawiki dictionary serves two purposes. People with editing rights can edit the dictionary easily online and other people can access an up-to-date dictionary for Uralic languages for free. The Mediawiki infrastructure also allows for variation in the extent of editing rights. This means that, while there are two different kinds of user groups whose needs have to be taken into account, there is also a possibility for monitored editing by a larger and larger contributing group of users.

For the editors, the system gives more information about each word in the page edit mode. This includes all sorts of metadata that is useful for researchers but not so useful for normal dictionary users. The reason why such information is not presented to everyone is that having too much irrelevant information would only confuse the normal dictionary users. They are probably using the dictionary to find what a certain word means in another language and they are not so interested in, let's say, the source from which a word came to the dictionary.

Currently, regular users are presented with lemma's part-of-speech, translation to different languages, semantic information and etymology. In the future we would like to include morphological transducers in the dictionary so that the user could see with a click of a button all the possible inflections of a word. Also we would like to link audio files and add example sentences to the entries of the dictionary.

5 Conclusions

The tandem development of analyzer dictionaries in an environment already hosting terminology, word list and corpora resources is a challenging undertaking in itself. Coordinating synchronized editing between multiple input infrastructures takes our efforts to a bold extreme.

The challenge is allowing for documentation of information already available from research. We are providing for input from multiple environments, and thus rendering collaboration with a lower threshold. The establishment of a platform for direct documentation of new and old results and data means we are essentially pointing the way in future open-source work in the field.

We are finding our niche by drawing on documentational structuring from multiple language research projects and documentational needs. We recognize that our development must be available for work being done in other areas, as we will be providing an intermediate point linking transducer development, semantics, etymology and corpora annotation.

The synchronized editing interfaces, Mediawiki and XML, will bring us to a new phase in dictionary development at CSC. The presence of unanalyzed or minimally analyzed corpora for Uralic languages will provide a resource for morphosyntactic information. New methods will be applied for semiautomated integration of dictionary development and simultaneous corpora annotation.