Exploring Natural Language Processing Methods for Finno-Ugric Langages

Thierry Poibeau, Benjamin Fagard

To cite this version:

HAL Id: hal-01273769
https://hal.archives-ouvertes.fr/hal-01273769

Submitted on 13 Feb 2016

Exploring Natural Language Processing Methods for Finno-Ugric Langages

Thierry Poibeau and Benjamin Fagard
PSL Research University
CNRS and Ecole normale supérieure and U. Sorbonne Nouvelle
1 rue Maurice Arnoux, 92120 Montrouge, France
thierry.poibeau@ens.fr

Svetlana Toldova
National Research University “Higher School of Economics”
School of Linguistics
toldova@yandex.ru

January 14, 2016

Abstract

This paper presents some preliminary experiments concerning the automatic processing of Finno-Ugric languages with computers. We present symbolic methods as well as machine learning ones. Given the lack of corpora for some languages, we think finite-state transducers may sometimes be the best approach where only little data are available for learning. We also consider some machine learning approaches that could be valuably applied in this context, more specifically lightly supervised techniques involving a reduced sample of annotated data and larger amounts of non annotated data. Lastly we present the LAKME project that will explore new techniques for parsing morphology-rich languages.

1 Introduction

The Finno-Ugric language family includes more than 30 languages which are for a large part endangered [1]. Most of these languages are spoken by a declining number of speakers and there is thus a growing interest in documenting these languages. This includes the preservation, normalization and annotation of corpora, as well as the production of reference tools (lexicon, grammars) which can be re-used in various applications[1].

In this paper, we present some joint work between the Lattice laboratory at the Ecole normale supérieure in Paris and the National Research University of Moscow Higher School of Economics, to develop resources and techniques for Finno-Ugric languages. We explore symbolic methods (esp. finite-state transducers) as well as machine-learning ones, including unsupervised as well as supervised methods. We think there is a need to adapt methods to the problem since, given the language under consideration, texts can be available or not, and the same applies for dictionaries or annotated data. Lightly supervised methods (i.e. methods requiring a small sample of annotated data as well as larger amounts of non annotated data) are also considered since they seem especially relevant in our case.

The structure of the paper is as follows. We first consider briefly the corpora available in Moscow. We then detail some experiments we have done with finite-state transducers and with the morphological segmenter Morfessor. In the last section we describe the LAKME project, which aims at developing parsing techniques for morphology rich languages. We conclude with a few consideration on evaluation and some perspectives.

2 Available Corpora

The University of Moscow as well as the National Research University "Higher School of Economics" conduct regular field work campaigns concerning Finno-Ugric languages spoken in Russia. The data collected (mostly audio data which are then transcribed and analyzed) concern Mari, Komi, Udmurt, Khanty, Erzya and Moksha, among others.

Once transcribed, these data are available as raw text (sometimes with some annotation using the SIL format, see `http://www.sil.org`) but automatic tools would be very useful to assist linguists in this process. Our goal is thus to enrich these data with linguistic annotation so as to make them more visible and more specifically easier to use for researchers interested in a specific linguistic phenomenon.

[1] From this point of view, we share the same goal as several other projects. See, among others, the FinUgRevita project, described at `http://www.ieas-szeged.hu/finugrevita/`

3 Lexical Analysis through Finite-state transducers

Finite-state transducers (FSTs) are widely available and different implementations exist for the easy and quick development of efficient natural language processing systems. One example of such a toolbox for NLP applications is the Unitex platform developed at the University of Marne-la-Vallée in France[2]. Compared to other toolboxes, Unitex includes a graphical interface that makes it easy for the end user to develop his/her own resources without advanced skills in computer science. This toolbox includes resources for various languages including Finnish: resources for Finnish have been developed at the University of Caen[3]. Unitex is provided with a LGPL license, which means the software is open source and can be used in various contexts without restriction (academic as well as industrial contexts).

It is well known that finite-state transducers are especially efficient for word processing as well as for the recognition of local syntactic patterns. Thanks to FSTs it is possible to describe the lexicon of inflected forms of a language based on a list of stems and declension paradigms in a very compact way. Unitex allows such an implementation. Once compiled, the system produces a formal lexical analysis of all kinds of linguistic units (words as well as compounds and idioms), along with relevant information attached to the lexical forms. The current resources[4] cover more than 32.000 nouns (more than 800.000 surface forms) and 16.000 verbs (more than 7 millions surface forms).

Beyond lexical analysis, a typical application is the automatic recognition of local sequences of texts. A typical example is named entity recognition, which includes the recognition of person names, location names as well as dates and more generally any semantic pieces of information relevant for a given application. We have presented in 2003 the implementation of such a system for a dozen languages, including Finnish[5] [4]. The idea is now to address less visible Finno-Ugric languages.

FSTs are interesting in that they make it possible to describe a grammar through a collection of readable graphs. The description is generally compact since the formalism is recursive: a graph can include different subgraphs, as shown on figure 1, where the grey box refers to a subgraph called dynamically.

The drawback of FSTs is the time required to write a high-coverage grammar[6] as

[2] `http://www-igm.univ-mlv.fr/~unitex/`. See also Omorfi for a similar toolbox [2].

[3] `http://www.unicaen.fr/ufr/homme/linguistique/ressources/finnois/`

[4] Available on the web site of the University of Caen, see the previous footnote.

[5] The implementation was then made with Intex [3], which is no longer maintained. A transfer to another FST toolbox like Unitex would be quite straightforward.

[6] Time spent to write a grammar is hard to evaluate and d largely epends on the language under con-

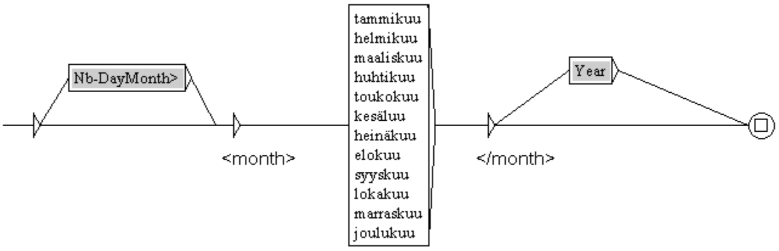


Figure 1: A Unitex graph (here an extract of a grammar recognizing dates in Finnish)

well as the maintenance of such a collection of graph, when its size expands. Machine learning techniques are known to generally give better results for a variety of tasks nowadays (especially morphosyntactic analysis) but it should also be noted that FSTs still provide fast and efficient implementations for a number of local linguistic phenomena, even with only limited data available for training. We thus think that FSTs remain interesting for endangered languages.

4 Automatic morphological segmentation using Morfessor

We have investigated the automatic segmentation of Moksha words with the Morfessor 2.0 software (see `http://www.cis.hut.fi/projects/morpho/` and `https://github.com/aalto-speech/morfessor`) [6, 7]. Morfessor uses raw text data and machine learning methods to find words segmentation in natural language. Morfessor can use unsupervised or semi-supervised methods, we tested both.

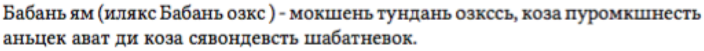
Our training corpus is composed of an extract of the mokshen pravda and an extract of the wikidumps of the Moksha wikipedia (`https://mdf.wikipedia.org/`). We have counted word frequencies for words written in cyrillic alphabet, in both sets of texts, leading to a word list of 120759 types (for 1352317 tokens).

For our first experiments, we used a test file is composed of only 16 sentences (183 words) from a wiki entry. For the semi-supervised approach we provided to

sideration and on other resources available. For example, [5] mention six months for a language and then 2 weeks for another language closely related to the first one (with a coverage between 85 and 92%, precision between 95 and 98% and recall between 57 and 85%, depending on the language under consideration).

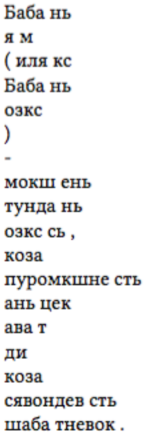
morfessor-train an annotation file with manually segmented words taken from a dictionary. The main drawback of this data set is that it is composed of non flexionnal forms but we are currently preparing a reference corpus of inflected forms so as to get more relevant results.

Here is for example a sentence in Moksha from our corpus:



Бабань ям (илякс Бабань озкс) - мокшень тундань озкссь, коза пуромкшнесть
аньцек ават ди коза сявондевсть шабатневок.

And here the automatic analysis proposed by Morfessor:



```
Баба нь
я м
( иля кс
Баба нь
озкс
)
-
мокш ень
тунда нь
озкс сь ,
коза
пуромкшне сть
ань цек
ава т
ди
коза
сявондев сть
шаба тневок .
```

Though these results are not optimal, we are confident that by providing more information to the system (so as to be able to guide the system with a semi-supervised approach – this is possible since Morfessor is provided with a purely unsupervised as well as with a lightly supervised mode) we will get more accurate results. These experiments are currently ongoing. A comparison could also be made with the results obtained with the Giellatekno Moksha analyser currently under development at the University of Tromsø[7].

[7] See `http://giellatekno.uit.no/cgi/index.mdf.eng.html`.

5 Machine learning approaches for Finno-Ugric languages: an overview of the LAKME project

LAKME is a project dedicated to the automatic production of linguistically annotated corpora. Textual corpora are nowadays largely available, including corpora for ancient as well as for under-resourced languages. However, from a linguistic point of view, these corpora are nothing if they are not enriched with linguistic information, allowing the researcher to go beyond purely "surface" patterns. At the same time, machine learning techniques and natural language processing (NLP) have made rapid progress, so that it is now possible to accurately analyse texts (at least at the morphosyntactic and syntactic levels). This project aims at developing new machine learning methods for text annotation. Targeted languages are Hebrew, French (esp. Medieval French) and Uralic languages.

For Uralic languages, the project involves researchers from the Lattice laboratory who develop machine learning methods while the National Research University Higher School of Economics from Moscow provides the data and some aid for the analysis.

Most of the developments in parsing have been done on English, for obvious reasons (importance of English as a communication language, existing evaluation campaigns, funding opportunities, etc.). However, an approach that is relevant for English may not be as efficient when applied to more diverse languages. The direct transfer of algorithms that are efficient for English to other languages has often led to unsatisfactory results, since language properties differ: at best, a simple adaptation from English leads to representation problems (e.g. when the model adopted for the English PennTreebank has been applied to Arabic, which is a free word order language), at worse it leads to annotation errors since the system makes wrong assumptions.

For morphology rich languages, it has been shown (see [8]) that it is mandatory to take into account language specific features. For example, in the case of Uralic languages, it is crucial to provide a fine-grained morphological analysis, capable of decomposing complex word beginnings and word endings, among other things. English or French are rather analytic, in that most of the relational information between words is supported by word position and specific relational words, esp. prepositions. This is not the case of most languages (like Hebrew, but also Arabic, Uralic languages or Japanese, to cite a few) and then, in this context, establishing a proper treatment of word morphology is both a complex and crucial task. This is why these languages are of prominent importance since English is highly unrepresentative from this point of view (English having a remarkably low degree of morphological complexity). Focusing the analysis on morphology-rich languages brings new challenges to the field and guarantees that the developed models are more adapted to language diversity.

A related topic concerns the treatment of unknown words. This is crucial for any parsing system but is even more important in the case of morphology-rich languages since most of the time the context does not give as many cues as in the case of analytic languages for word categorization.

6 Evaluation

Evaluation of natural language processing tools is an open research domain since evaluation must take into account the task, the domain and the context of development. We are nonetheless working on the development of gold standards for the different languages and tasks we are exploring, so that performances can be accurately measured. For most applications (for example part-of-speech tagging or named entity recognition) we think that relevant measures already exist (most of time, precision, recall and F-measure are relevant) and should also be used for Finno-Ugric languages whenever possible.

Using existing measures and open domain evaluation datasets allows one to compare results on a same task and sometimes across domains and/or languages. However, some tasks are clearly more difficult for morphology rich languages than for other languages with a low morphology complexity (as English). To address this issue, it could be interesting to be able to balance evaluation results with morphology complexity.

7 Conclusion

In this paper we have presented different experiments for the automatic analysis of Finno-Ugric languages. We have also given some details about future plans, more specifically through the description of the LAKME project. We are now working on practical experiments so as to get more detailed results soon on some Finno-Ugric languages from Russia (we are for example currently experimenting the automatic morphological segmentation of Moshka with Morfessor). We are especially open to collaboration since one of the objectives is to provide results for most languages, without duplicating similar work developed elsewhere.

8 Acknowledgements

We want to thank IDEX PSL (Paris Sciences et Lettres, ref. ANR-10-IDEX-0001-02 PSL) for supporting this research through the LAKME project. Svetlana Toldova has received a one-month invited professor fellowship at the Ecole normale supérieure in Paris, thanks to TransferS (laboratoire d'excellence, ANR-10-LABX-0099). Lastly, we would like to thank the reviewers for their insightful comments on the paper.

References

[1] Daniel Abondolo, editor. *The Uralic Languages.* Routledge, London, 1998.

[2] Tommi A. Pirinen. Omorfi—Free and open source morphological lexical database for Finnish. In *Proceedings of the 20th Nordic Conference of Computational Linguistics*, Vilnius, 2015.

[3] Max Silberztein. INTEX: a Finite State Transducer toolbox. *Theoretical Computer Science*, 231(1):33–46, 1998.

[4] Thierry Poibeau. The Multilingual Named Entity Recognition Framework. In *Proc. of the European Conference of the Association for Computational Linguistics (EACL 2003)*, pages 155–158, Budapest, 2003. Association for Computational Linguistics.

[5] Jonathan N. Washington, Ilnat Salimzyanov, and Francis M. Tyers. Finite-state morphological transducers for three Kypchak languages. In *Proceedings of the 9th Conference on Language Resources and Evaluation (LREC2014)*, Reykjavik, 2014.

[6] Mathias Creutz and Krista Lagus. Unsupervised discovery of morphemes. In *Proceedings of the Workshop on Morphological and Phonological Learning of ACL-02*, pages 21–30, Philadelphia, Pennsylvania, 2002.

[7] Sami Virpioja, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo. *Morfessor 2.0: Python Implementation and Extensions for Morfessor Baseline.* Aalto University publication series Science + Technology, 25/2013, ISBN 978-952-60-5501-5, Helsinki, 2013.

[8] Yoav Goldberg and Michael Elhadad. Word Segmentation, Unknown-word Resolution, and Morphological Agreement in a Hebrew Parsing System. *Computational Linguistics*, 9:121–160, 2013.