

Open Source Software and Natural Language Processing

Arbana Kadriu
SEE University
Tetove, Macedonia
a.kadriu@seeu.edu.mk

Natural Language Processing is one of the most rapid developing fields nowadays. But, unfortunately this is correct only for those languages that have commercial demand. For the other, minor languages this is not the case. So, they must rely on open-source toolkits and built-in software packages. For the most part of languages, pools of raw texts or even simple wordlists are the resources most urgently needed. In this paper, we try to analyze and bring some open source solutions that can help in building NLP applications for a wider spectrum of languages. These solutions include designing text corpora, lexical databases, implementing a complete morphological and sentence parser for a particular language, building language models etc.

Corpus data are, for many applications, the raw fuel of NLP, and/or the testbed on which an NLP application is evaluated. It is the main requirement for almost every kind of natural language processing, such as parsing, machine translation, information extraction, text mining, etc. So, since 1960, when one of the first instances of a text corpus was created (Brown corpus), many such corpuses were designed, together with the methods that were used in their production. In the last years, a new approach in designing textual corpora has been developed – crawling the text from the web. This new approach comes near the idea of having a heterogeneous and informal language of communication. And this also serves another purpose – it democratizes the way the linguistics work. This democratization is of special significance when dealing with small, non-market driven languages. In this paper, we will describe some of the open-source tools that produce web-based corpora for minor languages.

Corpus-based approaches to language have introduced new dimensions to linguistic description and to various applications by permitting some degree of automatic analysis of text. The most basic format used in displaying information about the linguistics elements in a corpus is generated by means of listing and counting. This field of NLP is known as statistical NLP. The CMU-Cambridge Statistical Language Modeling is a set of unix software tools designed to facilitate language modeling work in the research community.

One of the most explored fields of NLP is morphology. It is important because language is productive: in any given text we will encounter text words and word forms that we haven't seen before and that are not in a precompiled dictionary. The core task of computational morphology is to take a word as input and produce a morphological analysis for it. Morphotactics defines concatenation of the principal morphemes of a word, and it is typically described through finite state automata. But there are situations where the word formation process is not just joining of morphemes, such as assimilation, insertion, duplication, etc., and these are the situations where the phonological rules show

up. Phonological rules may apply and change the shape of morphs. Many linguists have modeled phonological rules, but it is considered that the most successful one is the model called *two-level morphology* (Koskenniemi, 1983). The two-level morphology model has been proved successful for formalising the morphologically of very different languages (English, German, Swedish, French, Spanish, Danish, Norwegian, Finnish, Russian, Turkish, Arab, Aymara, Swahili etc.). This system is used even for conversion between different writing systems.

PC-KIMMO is an open-source implementation for this formalism. It is of interest to computational linguists, descriptive linguists, and those developing natural language processing systems. The program is designed to generate (produce) and/or recognize (parse) words using a two-level model of word structure in which a word is represented as a correspondence between its lexical level form and its surface level form. The PC-KIMMO program is actually a shell program that serves as an interactive user interface to the primitive PC-KIMMO functions. These functions are available as a C-language source code library that can be included in a program written by the user.

Unsupervised learning refers to the computational task of making inferences (or acquiring knowledge) about the structure that lies behind some set of data without any direct access to that structure. In the case of unsupervised learning of morphology, and the possibilities of morpheme-combinations, for a set of words, based on *no* knowledge whatsoever of the language from which the words are drawn. *Linguistica* is a program which can be used to explore the unsupervised learning of natural language, with primary focus on morphology, which is to say, word-structure.

Another open-source toolkit for unsupervised learning is *Clog*. It is a simple but efficient first-order decision list learning system. Decision lists can be employed as a representation language for a wide range of tasks. *Clog* has been primarily developed with natural language applications in mind. So far, *Clog* has been successfully employed for morphology learning tasks.

Unsupervised learning can be used also to automatically learn two-level rules needed to implement two-level morphology formalism for a language. *Segmentize_and_learn_rules* is a set of perl scripts that make possible such a task.

Open-source toolkits and resources generate frames for research and data maintenance and this makes possible to overcome the isolationism of minor languages. Some other such toolkits and resources are: *TextMine*, *BootCat*, *NSP*, *DCGs*, *Wordnet* etc. And they play an important role to the advancement of language technology as a research area in general. This relates to the idea of free software that refers to freedom and progress, and not to the price.