Proceedings of the International Workshop on Intelligent Information Systems
IIS'2011, September 13-14, 2011, Chisinau, Republic of Moldova

Aspects of Stemmer and Lemmatiser Construction for Romanian Language Purposes

Elena Boian, Olga Palade, Mircea Petic

Abstract: In the article we describe the particularities of the process of stemmer and lemmatiser construction for Romanian language purposes. Starting with Romanian computational linguistic resources used for research description, we continue with the lemmatiser and stemmer particularities in the algorithm design.
Keywords: lemma, stem, lemmatizing, stemming, computational linguistic resources.

1 Introduction

For grammatical reasons, documents are going to use different forms of a word, such as *organize*, *organizes*, and *organizing*. Additionally, there are families of derivationally related words with similar meanings, such as *democracy*, *democratic*, and *democratization*. In many situations, it seems as if it would be useful for a search for one of these words to return documents that contain another word in the set. The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form.

However, the two words differ in their flavor. *Stemming* usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes. *Lemmatization* usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the *lemma*.

The aim of this article is to describe and to study the particularities of the process of stemmer and lemmatiser construction for Romanian language purposes.

In this order this paper is structured as follows. First the Romanian computational linguistic resources used for research is described. Then the

lemmatiser and stemmer particularities in the algorithm design are presented.

2 Database with Romanian linguistic information

In order to verify the correctness of the investigations on stemmer and lemmatiser construction there is a need of corresponding computational linguistic resource. For our purposes we have chosen the lexicon RRTLR[1] (*Resurse Reutilizabile pentru tehnologia limbii române*) that contains a database (DB) of linguistic resources at the word level and a set of DB management programs [1, 2].

In RRTLR these relations are determined by DB tables: table *words*, with 100.000 lemmas and table *word flexies* with 1.000.000 flexions. Flexions are formed on the base of a list of words. Every word has indicated its own flexion model. The sets of vocalic/consonantal alternations are defined for every single flexion model.

In addition DB contains sets of affixes in tables *prefixes* and *suffixes*, the derivatives and roots/stems from which they derive in table *deriv* consisting of information about 15.000 of derivatives and 8 thousand of roots/stems.

3 Lemmatiser

Unlike other lemmatizers for Romanian language lemma searching for a flexion form can be carried out from computational linguistic resources, information from the DB about the prefixed words, the searching flexion model algorithms for a arbitrar word in the case of being know the part of speech and for noun the gender.

The lemmatiser can derive lemma using a set of rules and a lexicon that express the relation between word forms and base forms [3].

For a flexion form that belong to table *word flexies*, it will be displayed all corresponding lemmas from table *words*, because the same lemma can have several entries. For example, for flexion form *capului* (neuter nouns: *cap - capuri*, *cap – capete*, masculine: *cap - capi)* respective from DB is to be displayed three lemmas *cap* – two neuter nouns and one masculine. On the base of table *words* from RRTLR and a

[1] http://imi201.math.md/elrr/

list of 1360 flexion forms from 4 domains there were found 84% of lemmas that belongs to DB.

For a prefixed flexion form that belongs to table *word deriv*, we will extract the root/stem without prefix. Then the root/stem is to be verified if it is present in table *word flexies.* If it belongs, all possible lemmas corresponding to table *words* are displayed. For example, for adjective *nefericiţii* with prefix *ne-* the corresponding lemma is *ne- fericit.*

There is another approach in the process of lemma getting, and it consists of the following steps [3]:

- in order to establish the flexion model a set of endings is constructed;
- basing on the set of endings the root is determined;
- vocalic/consonantal alternations are applied to the root;
- the final root is concatenated with the set of endings;
- the obtained flexion forms are registered in table *words flexies*;
- the obtained lemma is written in table *words*.

Taking into account the steps above a corresponding algorithm was elaborated and a program developed. The experiments, performed for a set of about 2000 base words, showed that in 87% of cases the lemma can be determined using the mechanism described above.

4 Stemming

Stemming is closely related to lemmatization. The difference is that a stemmer operates on a single word without knowledge of the context, and therefore cannot discriminate between words which have different meanings depending on part of speech.

Here we will study the following problem: how to find the root for a suffixed derivative and all its suffixes. For example for the word:`absolutizare – **absolut** iza re.`

Analysing the structure of derivatives, we came to a conclusion that it can be represented in the following way `root[s3][s2][s1]`, where `s1, s2, s3` are the possible suffixes and denotes that the suffixe may be missing. So, as output we will obtain a possible morphemic structure for a given word. The algorithm can determine the root only in the case the word has not flexion suffixes [4]. Basing on the facts above the algorithm consists of the following steps: finding of the possible suffixes `[s3][s2][s1]` and verifying their presence in DB.

The developed program based on the presented algorithm was tested on 300 of words from table *word_deriv*. The results correspond to 76% of a correct morphemic division.

5 Conclusion

Both stemming and lemmatization reduce inflectional forms and sometimes derivationally related forms of a word to a common base form.

Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes.

Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma.

Rather than using a stemmer, you can use a lemmatiser, a tool from Natural Language Processing which does full morphological analysis to accurately identify the lemma for each word.

References

[1] Ciubotaru, C., Cojocaru, S., Boian, E., Colesnicov, A., Malahova, L.,. Magariu, G., Petic, M. Verlan, T.,Burlaca, O., *Contribuții la proiectul "Roltech. Platformă pentru tehnologia limbii române: resurse, instrumente, interfețe"*, Lucrările Atelierului „Resurse Lingvistice şi Instrumente pentru prelucrarea limbii române", Ed. Universităţii „Alexandru Ioan Cuza", Iaşi, 2007, pp. 171-177.

[2] Boian E., O. Burlaca, C. Ciubotaru, S. Cojocaru, A. Colesnicov, G. Magariu, L. Malahov, M. Petic, T. Verlan, *"Application based on Reusable Linguistic Resources",* in Proceedings of ITSN-2010 "International Conference on Information Technologies, Systems and Networks 2010", Chișinau, 2010, pp. 54-63.

[3] S. Cojocaru, E. Boian. *Determination of inflexional group using P systems.* Computer Science Journal of Moldova, vol.18, no.1(52), 2010, pp.70-81.

[4] M. Petic. *Automatic derivational morphology contribution to Romanian lexical acquisition.* in: Natural Language Processing and its Application. Research in Computing Science, Mexico, vol. 46, 2010, pp. 67–78.

E. Boian, O. Palade, M. Petic

Institute of Mathematics and Computer Science of ASM
E-mails: lena@math.md, olga.palade@yahoo.com, mirsha@math.md