Aspects in developing of a text analizer for processing unstructured text data

Mircea Petic, Ecaterina Osoian

Computer Science and Mathematics Department, Alecu Russo Balti State University, Pushkin Str., 38, Bălți, 3100, REPUBLIC OF MOLDOVA, E-mail: petic.mircea@gmail.com

The article presents our approach in the elaboration of the system for processing unstructured text data in order to create a structured data output as computer linguistics resources using a lexicon of markers. First, a description of the research on the proposed topic, as well as its relation to the national and international level research is presented, being followed by the depiction of a useful to this particular research functionality - PoS Tagger for Romanian. A special section is dedicated to the algorithm to be used to elaborate our system. Finally, we describe several ways of marker lexicon completion by means of derivation.

Key words: computational linguistic resources, information technologies, linguistic Web services, unstructured data, linguistic markers.

Introduction

The development of information technologies, computer networks, and communication led to huge volumes of available information. This data is very difficult to process, as it is mostly unstructured, but it contains markers for social, cultural and security processes. Applying the means of text and sentiment analysis to extract and analyze this information is one of the objectives of our project. The project aims to elaborate the SoFTcrates tool, a software system for processing unstructured text data in order to create structured data output as computer linguistics resurces.

The described software system will be based on mechanisms of natural language processing and will create a great potential of interpretation of human-created text data in a systematic way with different purposes, like contextual analysis of the activity of software users from a statistical point of view or context and meaning detection of natural language. In the project we will explore and fill up the existing and published linguistic Web services, especially ones of Romanian origin.

The aim of this article is to present our approach in the elaboration of the system for processing unstructured text data in order to create structured data output as computer linguistics resources by means of lexicon of markers.

The paper is structured as follows: section 2 describes the correlation of the results obtained from the proposed topic with those nationally and internationally obtained by now; section 3 presents a tool that is to be very useful in our research, namely PoS Tagger for Romanian; in section 4 we describe the algorithm we will use to elaborate the system; section 5 includes the ways of marker lexicon completion by means of derivation.

I. State of the art

Most of the existing researches and applications of natural language processing are made in English. For this reason the European Commission has initiated a number of projects in support of the technologization of European languages other than the English language. This policy of promoting multilingualism through information technologies is continued, fact which can is proven by the priorities set out in the Framework Programme 7 (currently HORIZON 2020 Programme).

Analyzing various natural language processing applications one can find that, ultimately, the overwhelming majority of them are based on computational linguistic resources. The more voluminous and comprehensive are these resources, the more exact and qualitative results are obtained.

Romanian language begins to emerge as one of the significant languages in what concerns informatics resources and technologies applied to them. Therefore, the actual problem remains to automate the process of filling computational linguistic resources for Romanian.

The combination of methods from the field of information technology with the ones from linguistics would resolve the problem of computational linguistics, namely structured information retrieval of unstructured texts based on poetic texts.

Studying the history and the current situation in this problem for the proposed project based on results for Romanian, Russian, Italian, Spanish, French and Serbian lead to the conclusion of using existing Web services that work on morphosyntactic annotation.

II. Linguistic web services for the Romanian language.

Multiple Natural Language Processing tools have been and are currently developed in Romania, to fit the particular necessities of the local researchers and users of the products, based on general best practices in the domain. Below is a short description of the PoS Tagger for Romanian tool[25] developed by the Alexandru Ioan Cuza University from Iasi.

The application represents a language processing model based on both rule-based information extraction and statistical methods. The rules are used as constraints against ambiguity difficulties, creation of rules being facilitated by the use of Graphical Grammar Studio, an open-source software product which facilitates the

[25] http://nlptools.infoiasi.ro/WebPosRo/ - Simionescu Radu, UAIC Romanian Part of Speech Tagger, 2011

identification and matching of tokens and sequences that can be annotated as well as the simple tokens. PoS Tagger, as well as most of the other NLP utilities developed at UAIC, is published as a web service using the WSDL specification on top of the SOAP protocol [1].

III.The algorithm

The following example of algorithm represents the mechanism to be used for the further computer based identification of lines with Romanian chromatic words, using the Romanian Part of Speech Tagger web services[26] with the eventual completion of a dictionary of poetic meanings of colors.

The first step in the elaboration of the dictionary is establishing Chromatic concordances, which includes the extraction of the lines which contain a marker that is a word expressing a chromatic range.

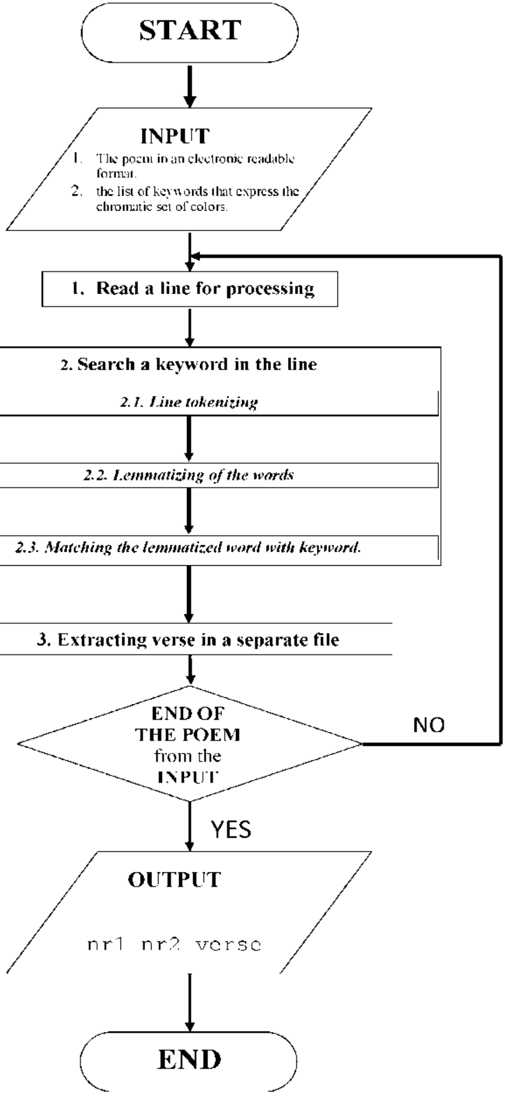
START
INPUT
1. The poem in an electronic readable format.
2. the list of keywords that express the chromatic set of colors.
1. Read a line for processing
2. Search a keyword in the line
2.1. Line tokenizing
2.2. Lemmatizing of the words
2.3. Matching the lemmatized word with keyword.
3. Extracting verse in a separate file
END OF THE POEM from the INPUT
NO
YES
OUTPUT
nr1 nr2 verse
END

Fig.1 Algorithm scheme

The lines will be then grouped and counted by keywords. In order to achieve it timely, automatic mechanisms may be used. The necessary condition for using the mechanisms is the *input* data (the poem) to be in an electronic readable format and to be supported by a list of keywords that express the chromatic set of colors. The *output* we will get is a file with lines like:

```
nr1 nr2 verse
```

where `nr1` represents the line number in the poem, `nr2` is the number that points to the first occurence in the verse of the keyword and `verse` represents the text of the verse line of the poem.

The natural way for automatic processing of these data in order to get the desired result is the following (Fig. 1):

1. Read from input a line to be processed;
2. Search for the word that matches the keyword from chromatic range;
3. Write the line that contains the keyword in a special file with the lines that contain keywords from corresponding chromatic range.

Below we will describe how each of the three steps listed above will be achieved.

Reading of a line for processing is made by using the standard procedures of reading from a file and text string processing.

Search in verse of a keyword occurs in several stages:

a. ***Verse division*** in separate words;
b. ***Lemmatizing*** of each word (getting lemma of the words);
c. ***Matching*** the lemmatized word with keyword.

To divide the verse into separate words, we will use special standard routines or those defined by the programmer to separate words in a verse.

Lemmatizing is probably the most difficult task in the algorithm. This part is facilitated by the use of previously developed publicly available web services. In our case, these are the facilities of PoS Tagger for Romanian, developed within the natural language processing research group at the Faculty of Computer Science of the “A. I. Cuza” University of Iasi. In this case, the procedures of verse division and lemmatizing will be merged because this web application offers the possibility to give the original data of a text and get an annotated (labeled) text with grammatical categories and word lemma. The existent WSDL services allow us to use the functionality in our particular software tool for our own research purposes [2].

The matching will take place with special subprograms that are available in the programming language that will be used.

Extracting verse in a separate file will be made by adding the containing keywords to the list of verses. For each keyword a separate file will be generated.

Even though the process of extraction of the lines containing words of a chromatic range is automated, there is need for some automatic validation of the performed work, in order to calculate the accuracy of the processing and improve it, especially when the Romanian sPoS Tagger is said by the creators to still have 3.3% of

[26] http://nlptools.infoiasi.ro/WebPosRo/ - Simionescu Radu, UAIC Romanian Part of Speech Tagger, 2011

inacurate output. That is where the analysis of the poem line acquisition process is be done by linguistic specialists.

IV.Automatic completion of markers lexicon

The particularities of the derivational morphology mechanisms help in lexical resources extension without any semantic information. Moreover, there are processing mechanisms similar for different languages spoken in Europe, namely English, French, Spanish, Russian, Romanian. The approaches and mechanisms presented in the paper have been studied on the examples from Romanian, but, in most of the of cases, they can be, more or less, applicable to other languages.

Affixes substitution

The ideea is inspired from Serbian derivational morphology [3], where the generated derivatives have predictible meanings, namely the gender modification in the case of suffix substitution, e. g., munci*tor* ↔ munci*toare* (eng. worker), and in the case of prefix substitution there is meaning change, e. g., *ante*belic ↔ *post*belic (eng. pre-war – after-war).

Affixes substitution is not an either Romanian or Serbian specific derivational morphology, but it is also characteristic for other European languages, e. g., Spanish (e. g., amortizar-amortizable, eng. to amortize-redeemable), French (e. g., revoir-prevoir, eng. revise-foresee), Russian (e. g., прочитать-дочитать, eng. read – read till the end) etc.

In general case for suffix substitution, let be x_1 a word of the form $x_1=\omega\alpha_1$ with the suffix α_1. After the substitution $\alpha_1\rightarrow\alpha_2$ we obtain the word $x_2=\omega\alpha_2$, e. g., corig*enţă*-corig*ent*. In the case of prefix substitution, let be x_1 a word of the form $x_1=\alpha_1\omega$, where α_1 is a prefix. After the substitution $\alpha_1\rightarrow\alpha_2$ we obtain the word $x_2=\alpha_2\omega$, where x_2 is the obtained derivative, e. g., *în*chide-*des*chide [4].

From the information above a new and original algorithm was developed which consists in examining the words in the lexicon and substitution of the affixes in those cases that correspond to the categories established by the above-mentioned rules.

Formal models

Formal models of derivation rules include a basis, from which derivative words are generated with a high degree of acuracy. A similar approach in derivational morphology is met in French language [5]. But when French system works with only 3 suffixes (-able,-ite,-is (er)) for which rules have been found, in the case of Romanian derivational morphology this study consist of 3 prefixes (ne-, re-, in-/im-) and 2 suffixes (-re,-iza).

- ***Rules for prefixes:***
  - ✓ *re-* $[\omega]_{inf} \longrightarrow [\mathrm{re}\ [\omega]_{inf}]_{inf}$
  - ✓ *ne-* $[\omega'\beta]_{adj} \longrightarrow [\mathrm{ne}\ [\omega'\beta]_{adj}]_{adj}$

    $\beta \in \{$*-tor, -bil, -os, -at, -it, -ut,-ind, -înd* $\}$
  - ✓ *in-/im-*$=\gamma$ $[\omega'\beta]_{adj} \longrightarrow [\gamma[\omega'\beta]_{adj}]_{adj}$

    $\beta \in \{$*-bil, -ent, -ant*$\}$
- ***Rules for suffixes:***
  - ✓ -*re* $[\omega]_{inf} \longrightarrow [[\omega]_{inf}\,\mathrm{re}]_{subst}$
  - ✓ -iza $[\omega'\beta\alpha]_{adj} \longrightarrow [[\omega'\beta]_{adj}\,\mathrm{iza}]_{inf}$

Derivatives projection

The projection of derivatives represents a method of word formation of the prefixed words from the suffixed words of the same root. According to Spanish researchers, the Spanish verb *amortizar* can be derived with the prefix *des*- obtaining *desamortizar*. Also, *amortizar* can be derived with suffixes *–cion* and *–able*. So, the derivative with prefix *des*- can derive with the suffixes *–cion* and *–able*. The hypothesis is that derivatives can inherit/project the derivatives with suffixes of the stem whose the prefixation was realized [6]. This method is not exclusively Spanish, but it can be applied to other languages; e. g., in English from the root *read* one can form derivatives readable and unread, therefore, it is possible to form the derivative unreadable.

Generalising the above noted, we conclude that it is possible to formally present the mechanism for Romanian derivational morphology. Let us consider a Romanian word ω, α - its prefix and β - its suffix. Then, the following relation is valid [4]:

$$(\omega\rightarrow\alpha\omega)\wedge(\omega\rightarrow\omega\beta)\Rightarrow(\omega\rightarrow\alpha\omega\beta),$$

for example, (a lucra → a *pre*lucra) ∧ (a lucra → lucr(a)ă*tor*) ⇒ (a lucra → *pre*lucr(a)ă*tor*);

$$(\omega\rightarrow\alpha\omega)\wedge(\omega\rightarrow\alpha\omega\beta)\Rightarrow(\omega\rightarrow\omega\beta),$$

for example, (a capitula→ recapitula) ∧ (a capitula →recapitulaţie) ⇒ (a capitula→capitulaţie)

$$(\omega\rightarrow\alpha\omega\beta)\wedge(\omega\rightarrow\omega\beta)\Rightarrow(\omega\rightarrow\alpha\omega),$$

for example, (a centraliza → *des*centraliza*tor*) ∧ (a centraliza → centraliza*tor*) ⇒ (a centraliza → *des*centraliza);

Examining the words in the lexicon and verifying them in correspondence with the relations above, a new and original algorithm has been developed that generates derivatives by affixes projection.

Derivational constraints

Where there is no clear model, according to which it would be possible to generate derivatives, some preconditions will appear, called derivational constraints. The most common derivational constraints are: parts of speech, inflection classes, affixes, changes that take place in the case of derivation, the letters preceding/succeding prefixes/suffixes. So, derivational constraints represent some schemes with several parameters that reduce the class roots and affixes in order to form derivatives. E. g. functions of the form:

$$f\colon \{\mathrm{wrd, pos, mod, sla, fgw, mvca}\} \rightarrow \mathrm{derivative}$$

where *wrd* is a word to derivate, *pos* - part of speech of *wrd*, *mod* - model of derivation, *sla* - the set of letters to which the affix is attached, *fgw* - flection group of *wrd*, *mvca* - modifications and vocalic or consonant alternations [4].

Examining the words in the lexicon and verifying them in correspondence with the relations above, an algorithm of derivatives generation by derivational constraints has been developed [7].

As examples of this method of derivatives generating, automatic derivation of words with the prefix *des-* and suffixes *-bil* and *-ime* can be analized.

f: {*a spinteca*, *verb*, des<verb>, ...*s*..., V14, evitarea dublării consoanei } → *de*(*s*)spinteca.

f: {*a programa*, *verb*, <verb>bil-itate, ...*a*..., V201, ... } → programa*bil*itate

f: {*crud*, *adjectiv*, des<adjectiv>, ..., A3, alternanța consonantică d - z } → cru(d)z*ime*.

Therefore, derivational constraints necessary for the automatic generation process do not depend on just the affix type, but also on the value of the prefix or suffix, considering the fact that each language has its own particularities in the derivation of words.

Conclusion

SoFTcrates will represent a substantial contribution in the field of natural language processing software as a synthesis, intermediation and enhancement of existing instruments in the field. Therefore, it will allow a more coherent use of the described intelligent mechanisms, facilitating access to it for a bigger range of users and contributing to spreading of competences in this field. Last, but not least, it will facilitate more research activity in some least explored areas of natural language processing by means of information, inclusion and facilitation of access of students to the problems, resources and mechanisms that this domain includes.

The implementation of the software system will imply reciprocal support, collaboration and exchange among a wide group of specialists, including the creators of the project, specialists from within and outside the country, in order to achieve a more coherent use of methods, information and linguistic technologies for reaching the goal of the project. Young researchers will have the opportunity to design new software architectural models and implement them wisely in the scope of computational linguistics. The tool created will be helpful to different users, but especially to philologists in their work on literary text resources for research purposes.

Acknowledgements.

This article is carried out as part of the project ”Developing of a text processing system with heterogeneous structure” supported by Supreme Council for Science and Technological Development from Republic of Moldova.

References

1. Simionescu R. Hybrid POS Tagger. In: Proceedings of “Language Resources and Tools with Industrial Applications” Workshop (Eurolan 2011 summerschool), 2011.

2. Petic M., Raciula L., Computer Based Identification of Lines with Romanian Chromatic Words from Poems, In: Electrotechnic and Computer Systems Journal, № 13 (89), 2014, Section Systems of Artificial Intelligence, 2014, Odessa, pp. 114-119

3. Duško V., Krstev C. Derivational Morphology in a E-Dictionary of Serbian. In: Zygmunt Vetulani (ed.), Proceedings of the 2nd Language & Technology Conference. Poznan, Poland, 2005, p. 139-143.

4. Petic M. Lexical derivation approaches for functional extention of computational linguistic resources. In: Proceedings of the 8th International Conference “Linguistic Resources and Tools for processing of the Romanian language” 8-9 december 2011, 26-27 april 2012. Bucharest, Editura Universitatii “Alexandru Ioan Cuza” Iasi, pp. 29-38

5. Fiammetta N., Dal G. GéDériF: Automatic generation and analysis of morphologically constructed lexical resources. Second International Conference on Language Resources and Evaluation (LREC). Athens, Greece, May 31 – June 2, 2000, p. 1447–1454.

6. Santana O., Perez J., Carreras F. and Rodriges G. Suffixal and Prefixal Morpholexical Relationships of Spanish, Lecture Notes in Artificial Intelligence, Ed. Springer-Verlag, 2004, pp. 407-418.

7. Petic. M. Computational linguistic resources interoperabilit in automatic lexical derivation. In: Proceedings of the International Conference “Human, Computer and Communication” HCC2013 28-29 May 2013. Lviv, Ukraine, pp. 25-28.