A Text Processing Tool for the Romanian Language

Oana Frunza and **Diana Inkpen**
School of Information Technology and Engineering
University of Ottawa
Ottawa, ON, K1N 6N5, Canada
{ofrunza,diana}@site.uottawa.ca

David Nadeau
Institute for Information Technology
National Research Council of Canada *
Ottawa, ON, Canada
David.Nadeau@nrc-cnrc.gc.ca

Abstract

BALIE[1] is a multilingual text processing tool designed to support information extraction. In this paper we explain how we adapted it to work for the Romanian language. With this addition, the tool supports five languages: English, French, German, Spanish, and Romanian. The services offered by the tool are: language identification, tokenization, sentence boundary detection, and part-of–speech tagging. We also present evaluation and results for the four newly added components for the Romanian language.

1 Introduction

Multilingualism, once a major problem for many Natural Language Processing tasks, is now finding a solution in software available on the market. Tools that process and extract information from texts in different languages are developed to support automatic processing.

One of the tools that are able to extract basic information from an English, French, German, Spanish or Romanian text is BALIE (Nadeau, 2005)[1]. It is a Multilingual System, able to find and structure data from free-written texts. At the moment, it provides the following services:

- Language identification;
- Tokenization;
- Sentence boundary detection;
- Part-of–speech (POS) tagging.

A raw text processed by BALIE is transformed into a structured and rich list of tokens. See Appendix 1 for a short example. A longer example can be found at: http://balie.sourceforge.net/sampleoutput.xml

The main difference between BALIE and other competing software that provide the same services (e.g., Gate[2], Oak[3] or MinorThird[4]) is that BALIE is ***trainable***. For each task, except for tokenization, BALIE uses machine learning techniques to learn from a sample corpus; no handwritten rules are needed. The learning process is done using machine learning techniques, provided by the Weka system (Witten and Frank, 2000). In fact, the multilingual capabilities of BALIE distinguish it from other software as well (most of the systems that provide the services mentioned above are only for one language, most often English and they might perform only one task).

Our system, RO-BALIE[5], is an extension of BALIE. RO-BALIE adds services for the Romanian Language, and also contains improvements for all the languages in the system, regarding the low-level processing of texts and the way the system reads the input data. All files that BALIE and RO-BALIE use for training or testing must be in a UTF-8 encoding. The output of the system is richer in information than the original BALIE's output; it includes the guessed language, the number of tokens in the file and the number of each sentence. See Appendix 2 for an example of a short Romanian text.

The next sections of the paper will focus on describing each of the four modules for Romanian Language, with evaluation of their perform-

[1] BALIE is a Java open-source software issued under the GNU General Public License. It is hosted by SourceForge and it is available at http://balie.sourceforge.net

[2] http://gate.ac.uk

[3] http://nlp.cs.nyu.edu/oak

[4] http://minorthird.sourceforge.net

[5] RO-BALIE is available for download at http://www.site.uottawa.ca/~ofrunza/ RO-Balie/RO-Balie.html

ance. For discussion about the same modules in BALIE's original languages (no evaluation results yet), refer to Nadeau (2005).

2 Language Identification

The language identification task is thought to be a solved problem. There are a lot of tools, some available online, some commercial, that give very good results. The Lextech Language Identifier[6] system supports 260 languages in different character encodings, with an accuracy of almost 100% for texts of minimum 250 characters. The system will also give as a result the name of the languages that are most similar with the one that it was guessed. The method that they used is not mentioned and neither is the size of the training files.

Takei and Sogukpainar (2004) built a system that uses unigram frequencies for classifying 4 languages. They reported a 98% accuracy using a cosine vector comparison. No information about the size of the documents is provided.

BALIE deals with this task from a machine learning point of view, creating a Language Model for each of the supported languages. For each language, we used a corpus (approximately 50 files, several pages long) for training and around 28 files for testing. The learning process is done using n-grams (sequences of n characters). For now, we used bigrams and unigram frequencies.

The language identification module of our system is based on the work of Beesley (1999). We used a Naïve Bayes classifier. BALIE guesses the language of a tested text as being the language that has the highest probability. It will also provide the probability values for all languages.

The formula that the Naïve Bayes classifier uses to determine the probability of a new text being in one of the languages supported by the system is:

$$P(H \mid E) = P(E_1 \mid H)\, P(E_2 \mid H) \cdots P(E_n \mid H) / P(E)$$

where H is the hypothesis, one of the classes (one of the languages) and E is the set of attributes $E_1, E_2, \ldots, E_n$ used in the classification.

Based on this formula the system will predict the guessed language as the language that has the highest probability.

Integrating a new language to BALIE will require a corpus of files for the new language. A Romance East–European language, with a Slavic influence, Romanian is not a very easy language to learn. Romanian is in a way similar to Spanish, Italian, and French, and a language identification process on a short text can be easily "fooled" if not using the right features. Even though there are special diacritics in the alphabet of the Romanian language, we cannot rely only on them for Language Identification. What if there is a quote in an English sentence that uses only one word from the Romanian language that has a special character? This problem can appear in any other language that has specific characters. Also, special characters might not be used in a given text, that is, the texts are written without the diacritics for the special characters. For Romanian this happens frequently.

The training corpus for the Romanian Language contains texts from different fields (literature, history, science, medicine, etc.) collected from the Web. Some of the files have the characters with Romanian diacritics, while some do not.

2.1 Evaluation and Results

We evaluated BALIE on a set of 137 files of sizes from 2 Kb to 150 Kb with an average of 28 files per language. Table 1 presents the results of the classification between the five languages supported by the system.

Table1. Language Identification results

Language	Files Train	Files Test	Correctly classified	Accuracy
English	50	27	27	100%
French	50	26	25	96%
Spanish	50	25	25	100%
German	50	27	27	100%
Romanian	50	32	32	100%

The overall accuracy of the Language Identification task is: **99.25%**. The result can be improved by adding new training files to each language and trying different parameters for the n-grams and for the buffer size (the system uses a memory buffer to store the most frequent n-grams). Taking into consideration that BALIE is a Java system, the memory can be an issue.

The mistake that the system did in the above experiment is between two languages that are close: French and Romanian. The French file that is classified as Romanian does not have any French specific characters.

[6] http://www.languageidentifier.com

The market for the Language Identification tools is well-developed, but most of tools are commercial and are not able to perform other tasks than identifying the language of a text. BALIE is a system that provides the basic information needed for various natural language processing tasks.

3 Tokenization

Tokenization is the task of splitting a text into its token components and is an important task for any system that deals with texts. Discussions and debates on the way this task should be performed can be found for any language. Some examples of common choices are: to separate the English possessive mark from the main word; to handle the French word *aujourd'hui* differently than other words with apostrophe; to split German agglutinative words; to handle the hyphen in pronoun-verb inversions for French and Romanian, etc.

For the Romanian language, one issue on the tokenization is splitting or not on the dash. For example "socio-economic" should be considered two distinct tokens or just a single one? Since at this stage RO-BALIE does not deal with compounds and named entities recognition, we decided to split each compound token based on the "-". This way the compound word "socio-economic" will be transformed into three tokens. While there is room for debate if some words should be split on the dash, when it comes to pronoun-verb inversion, e.g., *iat-o*, the decision of splitting is the correct one to take, since the pronoun "o" is a token with a different part-of-speech. We also decided to split tokens in Romanian texts based on the slash mark. A token like "25/455" will be split into 25 / 455.

For all the languages in the system, the first basic rule of splitting the tokens of a text is based on the space characters. The way to split the internal punctuation from words is a relevant part of this task. The system treats as separate tokens the leading and trailing punctuation. For example *"(informație)"* will be split into five tokens: *", (, informație,) and ".*

3. 1 Evaluation and Results

To measure the performance of the RO-BALIE system on the tokenization task, we used a text of 904 tokens, from a Romanian newspaper.

Table2. Tokenization results

Tokens	Precision	Recall
904	99.5%	98.7%

Most of the errors were due to the noise in the data. For example "PRM.Cel" was considered a single token since there was no space between the period, mark of the end of the sentence and the beginning of the next sentence that started with "Cel".

It is easy to add more tokenization rules to the system, to improve the results, and to tune the tokenization in the manner needed for a particular task.

In future work we plan to learn the tokenization rules from a sample tokenized corpus; this way the system will be totally trainable.

4 Sentence Boundary Detection (SBD)

Determining the sentence boundaries in a text is an important task for many Natural Language Processing systems – machine translation, parsing, information extraction, summarization, etc. The performance of any of these applications can be improved if an accurate SBD system is used.

One of the best systems that perform the task of sentence detection is Palmer and Hearst's (1997) system. They report 98.5% and 98.9% accuracy on the Wall Street Journal (WSJ). Their system uses the context to determine a potential sentence mark (the POS tags of six words before the potential mark and six words after it).

Grefenstette and Tapanainen (1994) built a system that uses regular expressions and a list of the most frequent abbreviations. An accuracy of 99.7% was established for sentences that end in period.

Mikheev (2000) reported a 0.25% error rate on the Brown corpus and a 0.39% error rate on WSJ with a system that uses the POS tags of the potential token and the two POS of the previous tokens.

A system that does not use the POS of the tokens is the one of Reynar and Ratnaparkhi (1997), based on a Maximum Entropy model. The model obtained an accuracy of 98.8% on the WSJ.

The Bondec system (Wang and Huang, 2003) incorporates a rule-based system with an 86.81% F-measure a Hidden Markov Model with a 92.92% F-measure, and a Maximum Entropy model with a 98.38% F-measure.

RO-BALIE (the same as BALIE) is a system that learns the SBD task. For the training part, we used a small corpus of 106 hand-tagged English sentences. We used the WEKA tool with J48 (Decision Tree) classifier. Our system does not need the POS of the words in the context from which we are performing the learning task.

RO-BALIE uses the following information from the context as features for the WEKA classifier:

- the token that is the beginning of the sentence;
- the previous token of the candidate sentence boundary;
- the candidate for the sentence boundary;
- the next token after the candidate.

The values that can be assigned to the features are:

- Period
- Period Like (? and !)
- Open Quote
- Close Quote
- Other Punctuation
- New Line
- New Line with all the tokens in the previous sentence in capital letter
- Capital word
- Digit
- Abbreviation
- Other word
- Null

Based on the selected features and on the possible values that can be assigned, the classifier will decide, using the learned model, if the candidate token is a sentence boundary or not.

Our system needs a list of abbreviation specific to each language. RO-BALIE uses a list of 510 abbreviations for Romanian. This number is not a small one, compared to the abbreviation lists for the other languages in BALIE and also compared to other systems for different languages that use similar types of lists (around 250-300 abbreviations for the English language).

The user can adjust the numbers of features (more context tokens), as well as the values of the features in order to increase the performance of the task.

4.1 Evaluation and Results

We evaluated the performance of our SBD module on the *Orwell's 1984* novel, both on English and on Romanian, from the MULTEXT-EAST project (Erjavec *et al.*, 1996). The learning process was done using the English text only, and the evaluation was done on the two languages, because we wanted to see how well the knowledge transfer performs.

Table3. Sentence Boundary Detection results

Text	**Accuracy**	**Precision**	**Recall**
Romanian	97%	92%	71%
English	97.5%	96.5%	82%

The results in Table 3 show for the SBD task are worse for the Romanian part of aligned corpus. In order to improve the performance, in future work we plan to train the SBD task for Romanian, rather than using the one trained for English.

The performance can be also be improved by training on a bigger corpus since the corpus that we used was really small.

5 Part-of-Speech Tagging

Knowing the part of speech of the words in a text is important for many NLP systems. Some of the most commonly used POS taggers are: the Brill Tagger (a transformation-based rule tagger) (Brill, 1992), and the TreeTagger (probabilistic) (Schmid, 1994). Most of the taggers that are available are only for English texts.

Our system uses the language-independent probabilistic tagger QTAG[7]. The corpus that we used for training QTAG is the ROCO corpus, a collection of 40 million words of newspaper articles from a Romanian newspaper, collected on the Web over a three years period (1999-2002). The corpus was tokenized and part-of-speech tagged with the RACAI's tools (Tufiş, 1999). It is estimated that the annotations are 98% accurate. The corpus also has named entities tagged, e.g., 1_aprilie_1999/Y, Evenimentul_Zilei_Online/NP.

To be able to perform the training part of the QTAG system, a preprocessing of the corpus was needed. We decided to split each compound-tagged token into the constituent tokens, and then assign the correct POS for each one. For example "1_aprilie_1999/Y" becomes "1 M aprilie NN 1999 M".

RO-BALIE has a tagset of 14 tags for POSs and 30 tags for punctuations. To be able to use

[7] http://www.english.bham.ac.uk/staff/omason/software/qtag.html

the corpus we also had to map the tagset that ROCO has to the tagset that our system uses.

Some of the tags that RO-BALIE uses are:

- Noun NN
- Adjective ADJ
- Verb VB
- Adverb ADV
- Particle PART
- Pronoun PR
- Number NO
- Preposition PREP
- Open Bracket OP
- Dash DS
- Slash SL
- ………..

The number of part-of-speech tags that the system uses is not very big, but it captures the main information needed by most of the NLP tasks. BALIE's tagset is the same for all five languages. Each language will map the tags from the original training corpus to the one that the system uses. The number of the tags is also adjustable, and the user can add as many tags as she/he wishes in order to perform the task as accurate as possible.

5.1 Evaluation and Results

We were able to train the tagger on a corpus of 25 million words (from the ROCO corpus) with a system that has 1.5G memory. Due to the memory limitations and long running time, we could not test on this model. We trained a model on 2.5 millions words.

We performed the evaluation of the tagger on a Romanian file containing only 13,425 tokens from the ROCO corpus, due to the same issue of memory and Java/QTAG limitations.

Table4. Part-of-Speech Tagging results

Train Corpus	Test Corpus	Accuracy
2.5 mil words	13,425 words	95.3%

Tufiş and Mason (1998) reported an accuracy of the QTAG tagger for the Romanian part of the translated *1984* novel of 97.82%, and for *The Republic* novel 96.10%. The novels have each of them around 100,000 words. They performed the training task on 90% (90,000 words) of the corpus and reported the results on the other 10% (10,000 words). Their system was specially designed and adapted for Romanian language using a tag set of 79 tags for parts-of-speech and 10 tags for punctuation.

The results that we had for our part-of-speech task are a bit more modest. This could be due to our smaller tagset and to the fact that our large training corpus is not 100% accurate.

We also ran experiments on the *Orwell's 1984* novel, in order to have a more accurate training set. We trained the tagger on 90% of the corpus and tested on 10%; we obtained 91.5% accuracy. The different result compared to Tufiş and Mason (1998) could again be caused by the different tagset.

In future work we plan to train the tagger to tag with both the original tagset of the ROCO corpus and BALIE's reduced tagset.

6 Related Work for Romanian

There is a considerable amount of research done toward the automatic processing of Romanian language. We highlight here some of the main directions. Some of these research projects were implemented in the context of multilingual projects, such as MULTEXT-East (Erjavec, 2004) (Erjavec *et al.*, 1996).

Several papers present morphological analyzers for Romanian (Tufiş 1997) (Vuşcan 1997). They were applied to spell checkers (Peev *et al.*, 1997) (Cojocaru, 1997) and to morphological taggers (Mason and Tufiş, 1997) (Tufiş, 1999).

Cucerzan and Yarowsky (2002) present a part-of-speech tagger for Romanian built by knowledge induction: it projects part-of-speech tags of English word to Romanian words by using a bilingual dictionary and other resources.

A Romanian wordnet was constructed as part of the BalkaNet project (Tufiş, 2004). Other applications for Romanian include: word sense disambiguation (Mihalcea *et al.*, 2004) (Şerban and Tătar, 2003) (Ide *et al.*, 2001), named entity recognition (Hamza *et al.*, 2003), text-to-speech synthesis (Ferencz *et al.*, 1998), and speech recognition (Boldea *et al.*, 1996).

The tools we present in this paper include a part-of-speech tagger for Romanian based on QTAG, similarly to (Mason and Tufiş, 1997). The differrence is that the tagset of Mason and Tufiş is much more fine-grained. Our tool also makes available the pre-processing modules: the tokenizer and the sentence boundary detector. Unlike most of the related work, our tool is easily available for download, and moreover it is open source. This means that its modules can be modified or replaced with other modules customized for particular applications. Another advantage is that the tool is multilingual (English, French, Spanish, German, and Romanian, for now). For example, if a document contains paragraphs in several languages, the language identification modules can be used to identify the language and then call the right part-of-speech tagger.

7 Conclusion and Future Work

We presented RO-BALIE, an extension of a multilingual system to include text processing services for the Romanian Language. We presented evaluation results for each module.

In future work we plan to modify the tokenization module to be able to learn rules from a tokenized corpus, rather than formulating the rules manually.

We also plan to expand the tool with new services (for all the languages in the system), starting with morphological analysis and named entity recognition. We are trying to add as much specific information as possible for each language separately in order to be able to perform better on each task.

Acknowledgements

Our research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), the University of Ottawa, and the National Research Council.

Appendix 1 A short example of BALIE's output for the input text *1. Introduction,* expressed in XML

```
<?xml version="1.0" ?>
<balie>
<tokenList>
<s>
<token type="2" pos="number"
canon="1">1</token>
<token type="1" pos="period"   canon=".">.</token>
<token type="2" pos="noun"
canon="introduction">Introduction</token>
</s>
</tokenList>
</balie>
```

Appendix 2 A short example of RO-BALIE's output for the input text *Apel tirziu si inutil NISTORESCU.*

```
<?xml version="1.0" ?>
<balie>
<Language ID="Romanian">
<tokenList>
<Tokens Count="896">
<s id="1">
<token type="2" pos="NN"
canon="apel">Apel</token>
<token type="2" pos="ADV" ca-
non="tirziu">tirziu</token>
<token type="2" pos="CJ" canon="si">si</token>
<token type="2" pos="NN"
canon="inutil">inutil</token>
<token type="2" pos="PN"
canon="nistorescu">NISTORESCU</token>
<token type="1" pos="PER" canon=".">.</token>
</s>
</Tokens>
</tokenList>
</Language>
</balie>
```

Note:
Token type =1 for punctuation
Token type =2 for words
Canonical form is only the lowercase of the word.

References

Eric Brill. 1992. A simple rule-based part-of-speech tagger, in *Proceedings of ANLP'92, 3rd Conference on Applied Natural Language Processing*, pp. 152-155, Trento, Italy.

Marian Boldea, Alin Doroga, Tiberiu Dumitrescu, and Maria Pescaru. 1996. Preliminaries to a Romanian speech database, In *Proccedings of ICSLP-1996*, pp. 1934-1937, Philadelphia, PA, USA

Svetlana Cojocaru. 1997. Romanian Lexicon: Tools, Implementation, Usage. In Dan Tufiş and Poul Andersen (eds.) *Recent Advanced in Romanian Language Technology,* Editura Academiei.

Silviu Cucerzan and David Yarowsky. 2002. Bootstrapping a Multilingual Part-of-speech Tagger in One Person-Day. In *Proceedings of the Sixth Conference on Natural Language Learning (CoNLL)*, Taipei, Taiwan.

Tomaž Erjavec. 2004. MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. In *Fourth International Conference on Language Resources and Evaluation,* Paris, France.

Tomaž Erjavec, Nancy Ide, Vladimır Petkevič, and Jean Véronis. 1996. MULTEXT-East: Multilingual text tools and corpora for Central and Eastern European languages. In *Proceedings of the First TELRI European Seminar: Language Resources* *for Language Technology*, pp. 87-98, Tihany, Hungary.

Attila Ferencz, Teodora Ratiu, Maria Ferencz, Tcillo Kovacs, Istvan Nagy, and Diana Zaiu. 1998. ROMVOX: Text-to-Speech Synthesis of Romanian, in Proceedings of the *Ninth International Workshop on Natural Language Generation*, Niagara-on-the-Lake, Ontario, Canada.

Gregory Grefenstette and Pasi Tapanainen. 1994. What is a word, what is a sentence? problems of tokenization, in *Proceedings of he 3rd International Conference on Computational Lexicography*, pp. 79-87, Budapest, Hungary.

Oana Hamza, Kalina Bontcheva, Diana Maynard, Valentin Tablan, Hamish Cunningham. 2003. Named Entity Recognition in Romanian using GATE. *RANLP 2003 Workshop on Information Extraction for Slavonic and Other Central and Eastern European Languages,* Borovets, Bulgaria.

Nancy Ide, Tomaz Erjavec and Dan Tufiş. 2001. Automatic Sense Tagging Using Parallel Corpora, in *Proceedings of NLPRS'2001*, pp. 212-219, Tokyo, Japan.

Oliver Mason and Dan Tufiş. 1997. Probabilistic Tagging in a Multi-lingual Environment: Making an English Tagger Understand Romanian. In *Proceedings of the Third European TELRI Seminar,* Montecatini, Italy.

Rada Mihalcea, Vivi Nastase, Timothy Chklovski, Doina Tătar, Dan Tufiş and Florentina Hristea. 2004. An Evaluation Exercise for Romanian Word Sense Disambiguation, in *Proceedings of ACL/SIGLEX Senseval-3*, Barcelona, Spain.

Andrei Mikheev. 2000. Tagging Sentence Boundaries. In *Proceedings of NAACL'2000*, Seattle, USA.

David Nadeau. 2005. Multilingual Information Extraction from Text with Machine Learning and Natural Language Processing Techniques. *Technical Report*, University of Ottawa. http://balie.sourceforge.net/dnadeau05balie.pdf

David D. Palmer and Marti A. Hearst. 1997. Adaptive Multilingual Sentence Boundary Disambiguation. *Computational Linguistics* 23(2).

Luciana Peev, Lidia Bibolar, and Jodal Endre. 1997. A Formalization Model of the Romanian Morphology. In Dan Tufiş and Poul Andersen (eds.) *Recent Advanced in Romanian Language Technology,* Editura Academiei.

Jeffrey C. Reynar and Adwait Ratnaparkhi. 1997. A Maximum Entropy Approach to Identifying Sentence Boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, Washington, D.C., USA.

Gabriela Şerban and Doina Tătar. 2003. Word Sense Disambiguation for Untagged Corpus: Application to Romanian Language. In *Proceedings of CICling 2003*, pp. 268-272, Mexico City, Mexico.

Helmut Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of International Conference on New Methods in Language Processing.*

Dan Tufiş and Oliver Mason. 1998 Tagging Romanian texts: A Case Study for QTAG, a Language Independent Probabilistic Tagger, *First International Conference on Language Resources and Evaluation*, Granada, Spain.

Dan Tufiş. 1999. Tiered Tagging and Combined Classifiers, in F. Jelinek and E. Nöth (eds) *Text, Speech and Dialogue*, Lecture Notes in Artificial Intelligence 1692, Springer.

Dan Tufiş (ed.) 2004. *Special Issue on BalkaNet. Romanian Journal of Information Science and Technology*, Volume 7, No. 1-2.

D. Tufiş, A.M. Barbu, V. Patraşcu, G. Rotariu, and C. Popescu. 1997. Corpora and Corpus-Based Morpho-Lexical Processing, in D. Tufiş, P. Andersen (eds.) *Recent Advances in Romanian Language Technology*, Editura Academiei.

Teodor Vuşcan, Emma Tamâianu, Sanda Cherata. 1997. SILEX - a Lexico-Morphological Software for Romanian. In Dan Tufiş and Poul Andersen (eds.) *Recent Advanced in Romanian Language Technology,* Editura Academiei.

Kenneth R. Beesley. 1999. Language Identifier: A Computer Program for Automatic Natural-Language Identification of On-line Text. *Languages at Crossroads: Proceedings of the 29th Annual Conference of the American Translators Association*, pp. 47-54.

Haoyi Wang and Yang Huang. 2003. Bondec – A Sentence Boundary Detector, http://nlp.stanford.edu/courses/cs224n/2003/fp/huangy/final_project.doc

H. Takei and I. Sogukpinar, 2004, Centroid-Based Language Identification Using Letter Feature Set. In the *Proceeding of CICLING 2004*, LNCS 2945, pp.640-648.

Ian H. Witten and Eibe Frank. 2000. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco, USA.