A learning-based Approach for Romanian Syllabification and Stress Assignment

Diana Balc, Anamaria Beleiu, Rodica Potolea and Camelia Lemnaru
Computer Science Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
{Rodica.Potolea, Camelia.Lemnaru}@cs.utcluj.ro

Abstract — **This paper tackles the Romanian syllabification and stress assignment problems, and proposes an efficient machine learning based solution. We show that by designing the appropriate feature sets for each specific problem, learning algorithms achieve satisfactory accuracy rates for both problems (~92% for syllabification, ~85% for stress assignment), even for relatively small training set sizes. We have found that unigram-based features are powerful enough to characterize these problems, and therefore the introduction of bi-gram or tri-gram features (often utilized in syllabification problems for other languages) is unnecessary.**

Keywords — syllabification; stress assignment; Romanian language; machine learning

I. INTRODUCTION

In this paper, we propose a machine learning based strategy for syllabification and stress assignment for words in documents written in the Romanian language. These tasks are important in many applications involving natural language processing such as text-to-speech processing, automatic speech recognition and letter-to-phoneme conversion.

Romanian grammar contains specific rules for syllabification; however the rules do not cover all of the words of the vocabulary. There are exceptions that need to be handled differently in order to identify hyphenation. For stress assignment on the other hand, the Romanian language does not offer straightforward and well defined rules to be applied on words. The stress is located only on vowels and in the majority of cases, on the last syllable. Moreover, stress is explicit only in the spoken language, not in the written one.

We developed a combined solution involving both tasks, but our experiments show that it is also possible to treat the problems independently. Our approach is to locate the position of the primary lexical stress and to use this information for predicting the syllable boundaries for a given word. The rest of the paper is structured as follows: Section 2 presents similar methods from the literature while Section 3 describes our proposed approach. Section 4 focuses on the evaluations performed with our method, and several interpretations are formulated. Finally, our conclusions are presented in section 5.

II. STATE OF THE ART

In the literature we can distinguish between two separate approaches for syllabification and stress assignment. The classical approach for the task of syllabification is based on known grammatical rules which are integrated into an algorithm for breaking a word into syllables. Such an approach is language dependent, since the rules for syllabification are different for each language. This method has been successfully used for automatic hyphenation – end-of-line word splitting and integrated into several text editing tools. Linguistic rules depending on the context have been gathered also for the task of stress assignment. Depending on the language, certain stress patterns can be identified [8]. However, there are words with irregular stress and patterns fail to identify the correct location of the stress.

A different approach is based on the deduction of rules from previously given examples enumerated in a dictionary. Therefore, the quality and number of words in the learning corpus are of great importance. Such methods are based on machine learning strategies and formulate the tasks of syllabification and stress assignment as a structured classification problem [1], [2]. The authors believe that there are patterns for both syllabification and stress assignment due to the regularities observed from correct examples. Although the number of n-grams may vary, the methods used in [1] and [2] use Hidden Markov Models and Support Vector Machines (SVM-HMM). This problem formulation requires a tagging scheme, for the relevant features to be marked. Positional tags (Not Boundary, Boundary – NB tags) and structural tags (Onset, Nucleus, and Coda) are employed. Each syllable is composed of a sequence of phones: a nucleus (vowel) proceeded by an onset (consonant) and followed by a coda (consonant). From the phonetic point of view, the nucleus and coda give the rhyme.

Probabilistic methods based on Conditional Random Fields have also been investigated in the literature for syllabification, which is modelled as a sequence learning problem in this case ([3], [4]).

Most of the studies performed on the syllabification and stress assignment problems by means of a dictionary-based approach have been applied on phonemes. Phonemes are a representation of a words' spelling which can be passed to a speech synthesizer component. However, the methods should be valid also for orthographic syllabification and stress assignment.

The goal of highlighting the location of the lexical stress might seem even more challenging, mainly because linguistic specialists argue that the Romanian stress system is predictable. In papers [8] and [9], the authors suggest that predicting Romanian stress is possible and formulate this task as a sequence tagging problem.

III. Method Overview

Syllable boundaries are not easy to identify, mainly due to several challenges in the Romanian language like the hiatus/diphthong ambiguity. Taking as an example the word “haină” - meaning “cloth” (noun) or “cruel” (adjective) - it can be split into syllables either as “h**a**i-nă” (as noun) or “ha-**i**-nă” (as adjective). In the first case, the group of vowels “ai” represents a hiatus, whereas in the second case it is a diphthong.

In addition, there are words with identical written form, but different syllabification and/or stress position. Homonyms are included in this broader category. Table I gives examples of other particularities given by the context in which the word is used. Another challenge is represented by the fact that words in the Romanian language contain diacritics. Table II gives information about the possible diacritics in the Romanian language and shows the corresponding chosen representation without marked symbols.

Correctly typed texts will contain these symbols, but we shall consider also the possibility that diacritics are omitted. Not having diacritics might decrease the accuracy of our method. In order to tackle this problem, we have investigated two different models – one that learns from words with diacritics and another that learns from words without them (replacing the specific diacritics with their corresponding non-diacritic symbol). Depending on whether tested words contain diacritic symbols, only one of the models is used.

We have performed a quantitative analysis on the “difficult” cases present in the Romanian language (i.e. words which may have different stress position and syllable boundaries).

TABLE I. EXCEPTION CASES

	Word	POS
Two accepted syllabification variants, same stress location	*i-gnór* (ignore) *ig-nór*	Verb (present)
Different part of speech and stress	*e-chi-pá* (prepare) *e-chí-pa* (team)	Verb (past) Noun (singular)
Different verbal tense and stress	*no-ti-fi-cắ* (*notified*) *no-tí-fi-că* (*notify*)	Verb (past) Verb (present)
Different word meaning and stress	*re-gíi* (*kings*) *ré-gii* (*administration*)	Noun (plural)
Different part of speech, stress and syllabification	*bi-blio-gra-fi-á* (*annotate*) *bi-bli-o-gra-fí-a* (*bibliography*)	Verb (present) Noun (singular)

TABLE II. DIACRITICS REPLACEMENT

â	Ă	□	ş	ţ
a	A	I	s	t

For this analysis we have employed perhaps the most complete dictionary which contains syllabification and stress information, *RoSyllabiDict* [5]. The dictionary consists of words and their syllabification variants together with lexical accents (in case there are more accepted variants for a given word, the dictionary also gives information about morphological aspects of words). It contains 525,534 inflected word forms, more than 65,000 lemmas, with their syllabification variants and accent information.

We performed an analysis on the exception cases. We identified approximately 15,000 exception examples. The great majority (~ 68%) of exceptions refer to the situation in which two syllabification variants are accepted as correct. 14.2% reflect different prominent syllables due to the part of speech, and 7.5% due to the verbal tense. Among the remaining, almost 10% are words whose syllabification and stress location are different.

In the literature, linguists have gathered a set of rules that apply to syllabification, based on consonant-vowel sequences. Generally, the rules for splitting a word into syllables can be divided into two categories: rules applied to sequences of consonants and vowels, respectively. The first category is easier to formalize and exceptions occur less frequently, whereas the rules dealing with vocalic sequences are prone to the appearance of false syllables. In the absence of stress information, the method cannot overcome the hiatus-diphthong ambiguity.

Moreover, due to the large number of exceptions from these rules, the performance of a pure rule-based implementation is limited. A hybrid approach would be preferred as it might better deal with the exceptions. Given the complexity of the syllabic structures, a data-driven approach was taken into consideration. Having a large amount of labelled data available, we employed a supervised solution at word level. A data driven approach offers several advantages, provided there is a sufficient quantity of training and validation data, which is our case. In figure 1, an overview of our data driven approach is illustrated; specific details about the model construction are presented in the following paragraphs.

The feature vector was defined in an iterative process, based on experimental validations. In order to transform the problem data into a format which is suitable for a regular classification process, we have transformed each word into a set of instances (with the size equal to the length of the word) and the class label being the break/no break decision (that is, whether the given letter represents the end of a syllable).

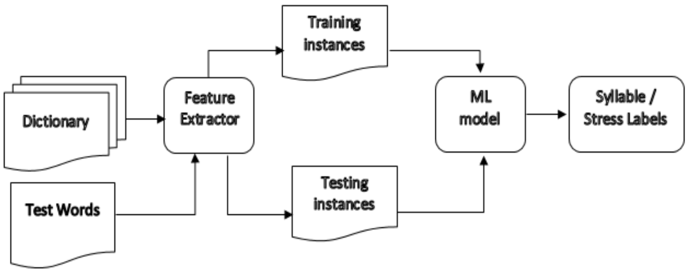

Fig. 1. Overview of the Method

We adopted the following formalization for the tasks of syllabification and stress assignment: each word letter is an instance described by a set of features consisting of a window of size ten in neighboring letters, five on each side of the current letter. Padding with a special character ("*") was needed for letters with smaller windows sizes (having less than five neighbors on each side). For the syllabification assignment, besides these features we added another Boolean position, marking whether the current letter is a vowel (1) or a consonant (0). The class attribute distinguishes which letter is a syllabic boundary (*yes* = boundary, *no* = not boundary).

In a second version of the feature vector, we considered stress information as input feature, i.e. we added the location of the main stress (marked with *1* = stressed, *0* = missing stress).

In a third version we have considered also the addition of bigrams to the feature set in our strategy. This implies that each instance is characterized by neighboring groups of two letters each, following the same strategy of five neighbors on the left and on the right side. We observed from our experiments that the processing time for training and testing increases significantly for this version of the feature vector. Moreover, considering the same size of training and testing sets for unigrams and bigrams, the boost in terms of accuracy is not notable. The model trained using bigrams would need much more instances in order to yield better results than in the case of unigrams. Consequently, we proceeded with features based on unigrams in our study.

Words containing more than one syllable are of interest during the training phase. Monosyllabic words don't offer valuable information for splitting a new word into syllables. As a consequence, monosyllabic words were not included in the training process. Table III gives examples of explicit instances obtained from the dictionary on the word "*înțelegi*". Our model is built using the second learning corpus mentioned above and used to predict whether a letter from a word is followed by another letter with or without a hyphen.

In a similar manner, training instances for the task of stress assignment are constructed. The difference lies in the fact that only vowels were considered as training/testing instances, since stress falls at vowel level (see Table IV). As a consequence, the resulting training sets have a smaller number of instances. Apart from the five neighboring letters on each side, the binary class attribute makes the distinction between stressed and not stressed letters (*yes* = stressed, *no* = not stressed).

TABLE III. EXAMPLE OF FEATURES EXTRACTED FOR SYLLABIFICATION OF THE WORD "ÎNȚELEGI"

Letter	*Features*	*Class*
Î	*0 1 * * * * * î n ț e l e*	*no*
N	*0 0 * * * * î n ț e l e g*	*yes*
Ț	*0 0 * * * î n ț e l e g i*	*no*
E	*0 1 * * î n ț e l e g i **	*yes*
L	*0 0 * î n ț e l e g i * **	*no*
E	*1 1 î n ț e l e g i * * **	*no*
G	*0 0 n ț e l e g i * * * **	*no*
I	*0 1 ț e l e g i * * * * **	*yes*

TABLE IV. EXAMPLE OF FEATURES FOR STRESS ASSIGNMENT OF THE WORD "ÎNȚELEGI"

Letter	*Features*	*Class*
î	** * * * * î n ț e l e*	*no*
e	** * î n ț e l e g i **	*no*
e	*î n ț e l e g i * * **	*yes*
i	*ț e l e g i * * * * **	*no*

IV. EXPERIMENTAL EVALUATIONS

Processing words from the dictionary according to the strategy described in Section III would result in 5.219.884 instances. In some cases, for a given word, there are multiple accepted syllabification variants. These include word forms derived with prefixes (10.311 examples) or obtained by composition. In the former case, the word "*inegal*" – meaning "*unequal*"- contain the prefix "*in*" and the word has two possible syllabifications: "*in-e-gal*" and "*i-ne-gal*". For the latter case, a word obtained by composition is "*despre*" which means „*about*". It has two different possible syllabifications: „*de-spre*" or „*des-pre*". We included in our training set only the second variant of these correct forms, the phonetic hyphenation.

After constructing our dataset, we proceeded to evaluating the performance of different classifiers. Our first choice was to build our model using a Support Vector Machine (SVM). SVMs have proven to work well for different natural language processing tasks in several languages [5]. We considered in our approach also decision trees, which scale well to a large number of examples of unseen data. Therefore, we use Random Forests (RFs) to create a different model for different cases and choose among them [6]. Another choice was Ada Boost, a member of the category of boosting algorithms [7]. The idea is to create an improved model iteratively by using previously built models and *learning* from the misclassified instances. Finally, we run experiments with Naïve Bayes, a well-known supervised learning algorithm applying Bayes' theorem. Naïve Bayes uses the assumption that attributes are conditionally independent of each other for a given class. In the evaluations performed, for all these classifiers we employed their corresponding Weka implementation.

In order to choose among the above mentioned classifiers, we generated five random training sets of words written with diacritics from the entire dictionary (following a uniform distribution). The number of words in each set was equal to 4.300 and in terms of number of instances their size ranged between 42.434 and 42.737. An evaluation set of 860 words (8.479 instances) containing different words with diacritics was also sampled from the dataset. The same evaluation set was used with each of the five training sets.

Table V illustrates the performance obtained in this scenario. The third column presents the values of correct classifications for at instance level (i.e. leter), with the best results obtained for RF and SVM. Based on these results, we decided to focus on the best two classifiers and analyze their efficiency at word level, on the same evaluation set.

TABLE V. CLASSIFIER COMPARISON FOR SYLLABIFICATION

Training set	Classifier	Accuracy (instance level)
Set 1	Random Forest	99,46%
	SMO	96,27 %
	Naive Bayes	87,03 %
	Ada Boost	80,92 %
Set 2	Random Forest	99,00 %
	SMO	96,04%
	Naive Bayes	87,09 %
	Ada Boost	80,92 %
Set 3	Random Forest	98,93 %
	SMO	96,02 %
	Naive Bayes	87,13 %
	Ada Boost	80,92 %
Set 4	Random Forest	98,93%
	SMO	95,94 %
	Naive Bayes	86,90 %
	Ada Boost	80,85 %
Set 5	Random Forest	98,93 %
	SMO	96,06 %
	Naive Bayes	87,05 %
	Ada Boost	80,92 %

We found that the word level average accuracy for RF was somewhere around 92%, with a standard deviation of ~2.3. On the other hand, the word accuracy of the SVM is poorer (around 70%). Therefore, subsequent evaluations focus on RF alone.

One might expect that the larger the training data, the better results we get. However, our experiments show that the proposed method achieves a good performance at a relatively small training set size, and that further adding training data is not necessary. Figure 2 illustrates the variation of the accuracy obtained at instance level using the RF algorithm, built using an increasing number of words in the training set for the task of syllabification. We can observe that with a growth from 2000 to 10000 words in the training set, the accuracy for syllable identification increases from 98.6% to 99.4% (words containing diacritics were employed for training and testing).

Having established that a training set size of about 4000 words achieves satisfactory results at this stage, we focused on assessing the performance stability of the built models, by assessing the performance of the same model on different evaluation sets (sampled uniformly from the dictionary, each containing ~860 words). For training we employed the same dataset (containing 4.295 words). Table VI presents the classification accuracies obtained at instance and word level. A mean of 98.95% was obtained at instance level, with a corresponding 90.32% accuracy at word level.

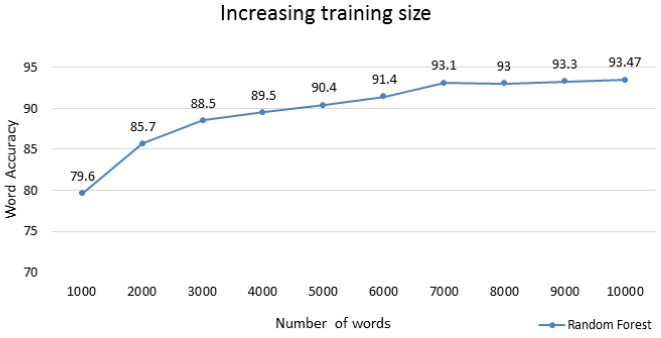

Fig. 2. Variation of the word accuracy with the training set size, for the RF classifier (syllabification task)

TABLE VI. RF MODEL EVALUATION USING FIVE DIFFERENT TEST SETS FOR SYLLABIFICATION – ACCURACY AT INSTANCE AND WORD LEVEL

Test set	Size	Accuracy (instance level)	Accuracy (word level)
1	8.552	98,86 %	89,768 %
2	8.525	98,97 %	90,233 %
3	8.541	98,74 %	88,838 %
4	8.544	98,81 %	88,605 %
5	8.489	99,41 %	94,187 %

For evaluating the task of stress assignment, we did not initially consider any syllabification information in the learning process (to have an assessment of how well the two tasks can be performed independently). Again, we employed five different training sets (containing around 13000 words) and the same test set, consisting of 860 words (8.479 instances). Table VII presents the accuracy values obtained at instance and word level for the task of stress assignment using RF, computed for words written with diacritics. The difference between the accuracy obtained at instance level and the one at word level is approximately 11 % - this difference is due to the fact that in some words more than one vowel is predicted as being stressed, whereas in others no stress is assigned. The results in Table VII present a relatively small standard deviation at instance level. However, the situation is different for the word level performance (standard deviation ~3.4).

TABLE VII. RF MODELS ACCURACY USING THE SAME TEST SET FOR STRESS ASSIGNMENT – INSTANCE AND WORD LEVEL

Train set	Size	Accuracy (instance level)	Accuracy (word level)
Set 1	58.434	96,37 %	87,79 %
Set 2	58.626	96,37 %	86,74 %
Set 3	58.207	96,26 %	86,04 %
Set 4	58.293	96,58 %	86,74 %
Set 5	58.288	93,53 %	77,56 %

Therefore, the performance for stress prediction seems to be worse than for syllabification, and also more variable (which makes it less reliable).

Next we evaluated the performance of the stress prediction approach on test sets of larger size, and using the same training set. The results are presented in Table VIII, where we include also the number of words with no stress predicted – which is quite high. To address this issue, we can enforce the rule according to which if no syllable is stressed in a word, the stress is automatically assigned to the last vowel in the word (if no syllabification info is available).

TABLE VIII. RF MODEL ACCURACIES (INSTANCE LEVEL) FOR DIFFERENT TEST SETS

Test set	No. test words	Test set size	Accuracy (instance level)	No. words w/o stress
1	1.100	4.860	96,04 %	143
2	2.200	9.720	95,72 %	318
3	3.300	14.410	96,12 %	389

Following our intuition, we attempted to use stress information to perform the syllabification task. Therefore, we included stress information in the feature vector for syllabification. The new feature vector, denoted by *Feature Vector 2* in Table IX, contains the location of the main stress as an additional feature following the strategy described in Section III. For this evaluation we employed a training set of size 46.667 (~ 4300 words). One can observe that the information about the stress location has not improved the accuracy of predicting syllable boundaries at word level. Therefore, even if our intuition was that stress assignment could be used for syllabification, the evaluation results indicate that it is easier to predict syllable boundaries than stress position – therefore – syllable boundaries information should be used in the stress assignment process.

TABLE IX. RF MODELS TESTED WITH THE TWO DIFFERENT FEATURE VECTORS (WITH AND WITHOUT STRESS INFORMATION) FOR THE TASK OF SYLLABIFICATION

Run	Accuracy (instance level)	Accuracy (word level)
Feature Vector 2	99.08 %	94.07 %
Feature Vector 1	99.48 %	95.58 %

V. CONCLUSIONS

The paper proposes a model capable of assigning stress to words written in the Romanian language and consecutively providing the syllabification of those words. Our results show that these two tasks can be viewed as separate and can be solved independently, but the result of performing syllabification could provide useful for stress assignment, and nit the other way around, as intuition would dictate us.

Our solution is modelled as a traditional classification task, in which each letter is modelled as a different instance. We have identified a set of features which capture relatively well the relation between the letter context and whether that letter is stressed and/or a syllable boundary. The algorithm which seems to be the most appropriate for these classification tasks is the Random Forest.

The accuracy of the method depends on the number of words present in the training set. We must mention that if we want to evaluate words with diacritics, we should have a train set of words with diacritics. One key observation from our experiments is that the number of words from the training set should not be smaller than 3,000 in order to ensure an accuracy of 95% at instance level. At word level, the results obtained for syllabification have a mean of ~92% and for stress assignment ~85%. That means the results obtained for the syllabification task following our approach are more accurate than the results for stress assignment, and we are currently focusing on improving stress assignment by employing syllable boundaries information. Also, our current work focuses on exploring context information in order to correctly identify homonyms and other exception cases, in order to achieve a good syllabification performance at sentence level.

ACKNOWLEDGEMENT

The research leading to these results has received partial funding from the Romanian Ministry of Education, under the grant agreement PN-II-PT-PCCA-2013-4-1660 (SWARA).

REFERENCES

[1] Bartlett, S., Kondrak, G., Cherry, C. 2008. Automatic Syllabification with Structured SVMs for Letter to Phoneme Conversion. 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL 2008: HLT), ACL, pp. 568-576

[2] Qing Dou, Shane Bergsma, Sittichai Jiampojamarn, and Grzegorz Kondrak. 2009. A ranking approach to stress prediction for letter-to-phoneme conversion. In Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing.

[3] Kseniya Rogova, Kris Demuynck, Dirk Van Compernolle. 2013. Automatic syllabification using segmental conditional random fields. In Computational Linguistics in the Netherlands Journal 3 (2013), 34-48.

[4] Liviu P. Dinu, Vlad Niculae, and Octavia-Maria Sulea. 2013. Romanian Syllabication Using Machine Learning. In Proc. of the 16th International Conference on Text, Speech and Dialogue, TSD 2013, pp 450–456.

[5] Ana-Maria Barbu, Romanian Lexical Data Bases: Inflected and Syllabic Forms Dictionaries. In Proceedings of the 6th International Conference on Language Resources and Evaluation, LREC 2008, pages 1937– 1941.

[6] Gérard Biau. 2012. Analysis of a random forests model, The Journal of Machine Learning Research, v.13 n.1, p.1063-1095.

[7] Robert E. Schapire, Yoav Freund, Petter Bartlett, Wee Sun Lee. 1998. Boosting the margin: A new explanation for the effectiveness of voting methods. In The Annals of Statistics, 26(5): 1651-1686.

[8] Ioana Chitoran, Alina Maria Ciobanu, Liviu Dinu, Vlad Niculae. Using a Machine Learning Model to Assess the Complexity of Stress Systems. Proc. of LREC 9, 2014, May 2014, Reykjavik, Iceland.

[9] Alina Maria Ciobanu, Anca Dinu, Liviu Dinu. 2014. Predicting Romanian Stress Assignment. In Proc. of the 14th Conf. of the European Chapter of the Association for Computational Linguistics, EACL 2014