A COMPARISON BETWEEN TRADITIONAL MACHINE LEARNING APPROACHES AND DEEP NEURAL NETWORKS FOR TEXT PROCESSING IN ROMANIAN

ADRIANA STAN, MIRCEA GIURGIU

Communications Department,
Technical University of Cluj-Napoca, Romania
Adriana.Stan@com.utcluj.ro
Mircea.Giurgiu@com.utcluj.ro

Abstract

This paper presents a comparison between traditional machine learning approaches (decision trees and multilayer perceptron) and the latest trend in artificial intelligence, deep neural networks for three separate tasks of text processing in Romanian. The tasks we examine are: lexical stress assignment, syllabification and phonetic transcription. The evaluation is performed on large manually transcribed lexicons and uses simple input features derived strictly from the orthographic form of the words. Results show that, depending on the task, the performance of each of the algorithms can vary, and that in some limited cases, the decision trees can outperform the deep neural networks.

Key words — decision trees, deep neural networks, lexical stress assignment, multilayer perceptron, phonetic transcription, Romanian, text processing, syllabification.

1. Introduction

Text processing is an essential part for most of the speech processing applications. In text-to-speech synthesis systems, for example, it is essential to know the correct syllabification, or the stress assignment of each word in the text input, so that the acoustic models can generate speech that is as natural as possible. Or, in automatic speech recognition systems, the language model decoders need to take into account the phonetic transcription of the words, so that a correct phrase is generated as output. But all text processing steps require large lexicons as training datasets.

In this respect, Romanian is an under resourced language (Trandabăț et al., 2012) and does not benefit from readily available, high quality language resources and systems. However, in the recent years more and more resources and tools have been developed and published as open-source or for academic use, such as (Barbu, 2008), (Toma et al., 2017), (Domokos et al., 2012), (Boroș et al., 2012), (Stan et al., 2017), (DEX online, 2018) or (Boroș et al., 2018). And with the availability of these resources, more and more studies and tools have been published, investigating either entire text processing systems, or only parts of them. The following paragraphs index some of the most important work with respect to the three text processing tasks we address in this paper: lexical stress assignment, syllabification and phonetic transcription. It is also worth mentioning a few papers which have introduced full systems, such as (Ungurean and

Burileanu, 2011) which addresses the diacritic restoration, text normalisation, syllabification, phonetic transcription and lexical stress positioning; (Stan et al., 2011) describes a complete text-to-speech synthesis system including the front-end text processing and acoustic modelling. (Boroș et al., 2012) and (Boroș et al, 2018) present the authors' work on a full text processing tool, as well as two newly developed Romanian speech corpora.

Lexical stress assignment was investigated by (Oancea and Bădulescu, 2002) by developing a set of rules which also use morphologic, lexical and phonetic information. (Ciobanu et al., 2014) and (Chitoran et al., 2014) use the consonant-vowel structure of the word and a cascaded model with averaged perceptron training to predict the syllable boundaries and lexical stress placement. (Balc et al., 2015) performed a series of experiments based on Random Forest algorithm for stress assignment.

The topic of syllabification in Romanian has been studied by (Toma et al., 2009) which presents a rule-based approach for syllabification and phonetic transcription. (Dinu et al., 2013) use sequence tagging, support vector machines and a set of rules. In (Balc et al, 2015) the authors perform a comparison of Random Forest, SMO, Naive Bayes and Ada Boost algorithms for syllabification. (Boroș et al, 2017) used modified decision trees for syllabification and phonetic transcription in order to improve the efficiency and speed of the two text processing steps as part of a complete system.

Phonetic transcription has seen more interest, as it is common to both the text-to-speech synthesis systems, as well as automatic speech recognition. In (Burileanu, 2002) the authors investigate the use of neural networks, a similar approach, based on ANNs, being discussed in (Jitcă et al, 2003) and (Domokos et al., 2011). On the other hand (Ordean et al, 2009) developed a large set of rules for the Romanian phonetic transcription and combined them with decision trees for exceptions. (Boroș et al, 2012) introduces a maximum entropy classifier and a purposely built specialized algorithm for same processing task. The authors of (Toma et al, 2013) compare 5 methods based on decision trees, neural networks, support vector machines, pronunciation by analogy and an expert system. A novel approach to the phonetic transcription task was presented in (Cucu et al, 2014) where grapheme-to-phoneme conversion is performed using statistical machine translation principles.

However, none of the previous cited works included the recent and revolutionising technique of deep learning (DL) for text processing. Therefore, in this paper we perform a comparison between two of the most common traditional machine learning methods used in the previous works and DL.

Deep Learning has seen a large increase in its use across many research topics (Zhang et al., 2018) and has revolutionised many of these fields, including computer vision, speech-enabled systems and information retrieval. Their main advantage over traditional machine learning algorithms is the fact that, due to their multiple hidden layers and training strategies, can learn non-linear high-dimensional partitions of the data spaces. But this comes at a cost, as multiple hidden layers require large amounts of training data. If this data is available, the accuracy of the networks is very high and their results are consistent across datasets.

Therefore, we also apply the deep learning models to the text processing tasks of lexical stress assignment, syllabification and phonetic transcription in Romanian and compare its results with two traditional machine learning methods: decision trees and multilayer perceptron.

The paper is structured as follows: Section 2 presents the text processing tasks and the selected algorithms. Section 3 introduces the performance analysis of the algorithms for each of the tasks, while Section 4 draws some conclusions and discusses future work.

2. *Method overview*

The most common tasks in speech-enabled applications refer to phonetic transcription, lexical stress assignment and syllabification. The following sections describe each of these tasks into more detail, highlighting the possible problems which may occur when basic rules or machine learning algorithms are used.

2.1. Lexical stress assignment

Stress assignment refers to the process of determining which syllable within a word is accentuated, or stressed. The assignment is in general inherent to a word's orthographic form, but within the limited set of homographs, the semantics also play an important role.

In Romanian, the stress pattern is rather variable and depends mostly on the etymology of a word. As a general rule of thumb, the stress can be considered to pertain to the penultimate syllable (Franzen and Horne, 1997) in a word. However, this does not hold true for most derivational and inflexional morphologic forms of the words, as well as to neologisms (e.g. *ac-ce'nt* - accent; *'a-ri-pă* - wing).

2.2. Syllabification

Syllabification is the process of segmenting a word into syllables. Syllables are considered to be the building block of a language and can determine its rhythm and complexity. There are 7 basic rules for syllabification in Romanian, which correspond to various vowel-consonant groups of letters. However, compound and foreign words do not adhere to these rules. There is also the very important issue of hiatus-diphthong ambiguity (Dinu et al., 2013) in Romanian (e.g. *bi-blio-gra-fi-á* - to annotate vs *bi-bli-o-gra-fi-a* - bibliography) .

2.3. Phonetic transcription

The orthographic form of most languages does not entirely reflect its phonology, or the pronunciation of the words. Translating the written form of a word into its acoustic realisation is essential in all speech-enabled applications, as well as in the systems which enable a user to learn a new language.

Romanian is a rather simple language in terms of letter-to-sound rules. Most of the letters have a direct correspondence with a single sound or phoneme[1]. The most problematic set of letters are the vowels: *e, i, o* and *u*, which can also be semi-vowels in diphthong and triphthong letter groups (e.g. ***oa**-**i**e* - sheep vs *chi-r**i**-e* - rent). Another set of problematic letters are *c* and *g* which are pronounced differently when followed by the letter groups *he* and *hi*. The letters *b* and *x* also has a different pronunciation depending on their context. If for the last two categories, rules can solve the issue, for the vowels, machine learning methods are required.

2.4. Algorithms

Given each of the tasks enumerated in the previous sections and the state-of-the-art presented in Section 1, we selected two traditional machine learning algorithms: decision trees and multilayer perceptron as being the most representative and accurate across the cited studies. Their performance on these tasks will be compared to that of the deep neural networks.

The classification and decision trees (Breiman et al., 1984) are a class of learning algorithms which make successive binary splits on the training dataset by using adequate attribute values as decisions in their nodes.

The multilayer perceptron (MLP) is a class of feedforward neural networks with at least 3 hidden layers and a non-linear activation function (Hastie et al., 2009). Their performance was state-of-the-art in the mid 80s, but have been replaced by the more simpler Support Vector Machines. MLPs are the predecessors of the more elaborate Deep Neural Networks.

Deep Neural Networks (DNNs) (Bengio et al., 2015) are a class of machine learning algorithms based on neural networks, where the number of hidden layers and nodes per layer is generally large. DNNs have been at the forefront of the machine learning field for the last decade. Their performance on various high complexity tasks have led to a great increase in their use across most of the supervised learning problems. Their main disadvantage has been the need for large amounts of training data, due to the large number of trainable parameters.

3. Results

3.1. Datasets

The evaluation of supervised algorithms requires large quantities of high-quality training data. And the tasks evaluated in this paper are no exception. Therefore, we selected large purposely built lexicons for each of the selected text processing tasks. The lexicons were manually created or checked, and have been previously used in similar evaluations.

The syllabification of approximately 507,000 words was extracted from the RoSyllabiDict lexicon (Barbu, 2008). For stress assignment, we used the DEX Online

[1] We are not taking into account allophones.

Database (DEX, 2000) which includes over 1,600,000 words and their inflected forms along with the stress labels. This large list of words was reduced to the subset of words also available in the syllabification corpus (507,000 words), for compute efficiency reasons. The MaRePhor phonetic dictionary (Toma et al, 2017) and its approximately 72,000 words represented the training data for our phonetic transcription experiments.

3.2. Features

As creating highly specialised input features can be computationally expensive, we resumed the use of integer encodings of the graphemes, augmented with simple positional and categorical features. The categorical features refer to the use of the vowel-consonant classification of the graphemes, while the positional features refer to the length of the word, as well as to the location of the current letter within the word as a forward percentage of the total number of the letters.

We used 2 strategies for defining the grapheme input features: left-right N-letter context, and entire word. The first strategy refers to extracting groups of letters of length 2N+1 from the current word. The letter situated in the centre of the group is the one for which the prediction is made. For consistency across the words, 0-padding was used. The length of the contexts or window was set to 3, 5, or 7 letters. In the entire-word prediction strategy, we created a 0-padded sequence of length equal to the length of the longest word in our lexicons (25 letters).

In lexical stress assignment, we did not construct input features which had a central consonant, as the stress is only assigned to vowels. The prediction for each input feature set stated if the current central vowel is stressed or not—for window contexts—, or which letter is stressed in the case of entire word prediction.

For syllabification, the output label marked if the current central letter is a syllable boundary or not. In the case of entire word prediction, the syllable boundaries were encoded as an integer obtained from the conversion of a binary number of 25 bits, where each binary digit encoded the syllable boundary presence at the respective index in the word.

The case of phonetic transcription is slightly different, as the entire-word strategy is hard to encode using simple classification methods. As a consequence, we do not report results for this strategy. Also, as only some of the letters in Romanian pose pronunciation problems, we evaluated the algorithms for the case of predicting all the letters, as well as only the problematic ones.

In all three text processing tasks' the use of letter types (consonant/vowel) was also evaluated. This means that along with the graphemes, the input features included the sequence of vowel-consonant types for each of the graphemes within the context window or word, respectively.

3.3. Implementation

All the evaluations were carried out using Python implementations of the algorithms. For the decision trees, we used the Python Scikit[2] DecisionTreeClassifier[3], with default parameters. The MLPs were implemented with the Python Scikit MLPClassifier, using three hidden layer of sizes equal to a halving progression: layer 1 - the length of the input features, layer 2 - half the length of the input features, layer 3 - a quarter of the length of the input features, and a ReLU activation function.

The Deep Neural Networks were implemented in Keras,[4] with a fully connected forward architecture of 3 hidden layers, similar to the MLP: layer 1 - half the size of the input features, layer 2 - a quarter of the size of the input feature, layer 3 - 10 nodes. In between each layer there was a dropout layer of 20% of the neurones. The output layer was a softmax. The training was performed on an nVIDIA 970 GPU. The training stopped after no increase in accuracy for 2 consecutive epochs. The network used an Adam optimiser (Kingma and Ba, 2014) and the loss was set to categorical cross-entropy.

3.4. Algorithm evaluation

The tables below present the accuracy of the 3 algorithms with respect to each of the 3 text processing tasks. We evaluated their accuracy and F-measure on a held-out randomly selected subset of 20% of the data. In the cases where the algorithms showed very similar accuracy across the evaluation sets, we report a 4 digit precision result. WIN3, WIN5 and WIN7 refer to the length of the context window.

3.5.Discussion

The results presented in the previous section show that the MLP is the worst performing algorithm. It might be true, however, that the tuning of its hyper-parameters could lead to a better performance. However, this was not the purpose of this study.

When comparing the decision trees with the DNNs it seems that their performance is quite similar, and in some limited cases, the decision trees marginally outperform DNNs.

However, the DNNs have proven their efficiency in the entire-word prediction task, where their accuracy is clearly above the one of the decision trees. It is also encouraging that the DNNs had an accuracy of at least 96% on each of the tasks, and in some cases, this accuracy almost reached 100%.

Table 1: Stress assignment evaluation in terms of accuracy and F-measure

Algorithm	Features	With letter type	Without letter type

[2] http://scikit-learn.org/stable/

[3] http://scikit-learn.org/stable/modules/tree.html

[4] https://keras.io/

Algorithm	Features	Accuracy	F-measure	Accuracy	F-measure
Decision Trees	WIN3	0.97	0.92	0.96	0.92
	WIN5	0.97	0.93	0.97	0.93
	WIN7	0.97	0.93	0.97	0.93
	Entire word	0.92	0.92	0.91	0.91
MLP	WIN3	0.88	0.72	0.84	0.56
	WIN5	0.92	0.79	0.87	0.65
	WIN7	0.92	0.82	0.87	0.66
	Entire word	0.79	0.74	0.52	0.46
DNNs	WIN3	0.96	0.92	0.96	0.90
	WIN5	0.98	0.95	0.97	0.95
	WIN7	0.98	0.95	0.98	0.95
	Entire word	0.96	0.96	0.95	0.95

Table 2: Syllabification evaluation in terms of accuracy and F-measure

Algorithm	Features	With letter type		Without letter type	
		Accuracy	**F-measure**	**Accuracy**	**F-measure**
Decision Trees	WIN3	0.99	0.99	0.99	0.99
	WIN5	0.99	0.99	0.99	0.98
	WIN7	0.99	0.99	0.99	0.98
	Entire word	0.86	0.86	0.67	0.67
MLP	WIN3	0.96	0.93	0.78	0.67
	WIN5	0.96	0.95	0.81	0.70
	WIN7	0.97	0.96	0.83	0.74
	Entire word	0.65	0.63	0.30	0.21
DNNs	WIN3	0.99	0.99	0.99	0.99
	WIN5	0.99	0.99	0.99	0.99
	WIN7	0.99	0.99	0.99	0.99
	Entire word	0.98	0.97	0.97	0.97

Table 3: Phonetic transcription - **problematic letters** - evaluation in terms of accuracy and F-measure

Algorithm	Features	With letter type		Without letter type	
		Accuracy	F-measure	Accuracy	F-measure
Decision Trees	WIN3	0.9882	0.9881	0.9859	0.9858
	WIN5	0.9877	0.9875	0.9837	0.9837
	WIN7	0.9868	0.9868	0.9825	0.9824
MLP	WIN3	0.86	0.84	0.80	0.73
	WIN5	0.90	0.88	0.86	0.80
	WIN7	0.93	0.92	0.87	0.83
DNNs	WIN3	0.99	0.99	0.98	0.98
	WIN5	0.99	0.99	0.99	0.99
	WIN7	0.99	0.99	0.99	0.99

Table 4: Phonetic transcription - **all letters** - evaluation in terms of accuracy and F-measure

Algorithm	Features	With letter type		Without letter type	
		Accuracy	F-measure	Accuracy	F-measure
Decision Trees	WIN3	0.9944	0.9944	0.9947	0.9947
	WIN5	0.9940	0.9939	0.9924	0.9924
	WIN7	0.9936	0.9937	0.9927	0.9926
MLP	WIN3	0.62	0.58	0.63	0.51
	WIN5	0.93	0.92	0.68	0.58
	WIN7	0.96	0.96	0.84	0.80
DNNs	WIN3	0.9958	0.9956	0.9957	0.9957
	WIN5	0.9963	0.9961	0.9963	0.9963
	WIN7	0.9960	0.9960	0.9950	0.9947

The letter type input features did not influence the results of the tasks or algorithms overall, but they did have a major impact in the syllabification task, especially for the MLP. This was to be expected, as most of the syllabification rules in Romanian use the vowel-consonant structure of the word.

In phonetic transcription, the accuracy of predicting only the problematic letters is comparable to the one where all the letters are predicted, even though these are only a small subset of the entire data. Predicting the entire set of letters would also impose additional computation expenses.

4. Conclusions

This paper presented a brief comparison between traditional machine learning algorithms: decision trees and multilayer perceptron, and the deep neural networks applied to three different text processing tasks in Romanian. The tasks included stress assignment, syllabification and phonetic transcription. The algorithms were compared in terms of accuracy and F-measure on several different input feature sets. The use of context windows versus entire word prediction strategy was evaluated, along with augmenting the input with positional and categorical features.

Results showed that the decision trees can achieve comparable accuracy with the DNNs, while the MLP is the least well performing algorithm out of the 3. Also, using additional features led to better results. The window length did not play such an important role in the accuracy, but window contexts performed better than the entire-word strategy.

As future work, in terms of traditional machine learning mechanisms, other algorithms could be evaluated, such as support vector machines. However, previous work showed that decision trees and multilayer perceptrons are best suited to these classification tasks.

Also, combining the features could lead to better accuracies of individual tasks. For example, the syllabification and stress assignment are important for phonetic transcription. The part-of-speech tagging is essential for diacritic restoration, and so on.

There is still a large amount of analysis to be performed with respect to the architecture of the MLP and DNNs and the tuning of their hyper-parameters. One other important issue is the use of sequence-to-sequence learning for all the tasks concurrently, as well as evaluating the speed of the algorithms when used in a complete processing system, and the overhead that they introduce in it.

Acknowledgement

This work was partially supported by a grant of the Romanian Ministry of Research and Innovation, PCCDI – UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0818/ 73PCCDI, within PNCDI III.

References

Barbu, A.M. (2008), Romanian Lexical Data Bases: Inflected and Syllabic Forms Dictionaries*, In Proc of Language Resources and Evaluation Conference,* Marrakech, Morocco.

Bengio, Y., LeCun, Y., Hinton, G. (2015), Deep Learning, *Nature* 521(7553): pp. 436–444.

Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A. (1984). Classification and Regression Trees. *Taylor & Francis.*

Boroș, T., Ștefănescu. D. and Ion, R. (2012). Bermuda, a data-driven tool for phonetic transcription of words, *In Proc. of the Workshop on Natural Language Processing for Improving Textual Accesibility,* Barcelona, Spain.

Boros, T., Dumitrescu, S.D. and Pipa, S. (2017). Fast and Accurate Decision Trees for Natural Language Processing Tasks. *In Proc. of Recent Advances in Natural Language Processing,* pp 103-110, Varna, Bulgaria.

Boroș, T., Dumitrescu, S.D. and Pais, V. (2018), Tools and resources for Romanian text-to-speech and speech-to-test applications, *arXiv, 1802.05583v1*

Burileanu, D. (2002). Basic Research and Implementation Decisions for a Text-to-speech Synthesis System in Romanian. *International Journal of Speech Technology,* no 5, pp. 211-225

Chitoran, I., Ciobanu, A.M., Dinu, L. and Niculae, V. (2014). Using a Machine Learning Model to Assess the Complexity of Stress Systems. *In Proc. of the Language Resources and Evaluation Conference,* Reykjavik, Iceland.

Ciobanu, A., Dinu, A., Dinu, L. (2014). Predicting Romanian Stress Assignment. *In Proc. of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp 64-68, Gothenburg, Sweden.

Cucu, H., Buzo, A., Besacier, L. and Burileanu, C. (2014). SMT-based ASR domain adaptation methods for under-resourced languages: Application to Romanian. *Speech Communication,* no 56, pp 195-212.

DEX online (2018), http://www.dexonline.ro

Dinu, L.P., Niculae, V. and Sulea, O.-M. (2013). Romanian Syllabication using Machine Learning. *In Proc. of the International Conference on Text, Speech and Dialogue,* pp. 450-456.

Domokos, J., Buza, O. and Toderean, G. (2011), Automated grapheme-to-phoneme conversion system for the Romanian Language, *In Proc. of the 6th Conference on Speech Technology and Human-Computer Dialogue,* Brasov, Romania.

Domokos, J., Buza, O. and Toderean, G. (2012), 100k+ Words, Machine-Readable, Pronunciation Dictionary for the Romanian Language, *In Proc of the 20th European Signal Processing Conference,* Bucharest, Romania.

Franzen, V. and Horne, M. (1997). Word stress in Romanian, *Working Papers 46,* pp 75-91.

Hastie, T., Tibshirani, R., Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. *Springer, New York, NY, 2009*.

Jitca, D., Teodorescu, H.N., Apopei, V. and Grigoras, F. (2003). An ANN-based method to improve the phonetic transcription and prosody modules of a TTS system for the Romanian language. *In Proc. of the 2nd Conference on Speech Technology and Human-Computer Dialogue.*

Kingma, D. P. and Ba, J. (2014). Adam: A Method for Stochastic Optimization. arXiv 1412.6980.

Oancea, E., Badulescu, A. (2002). Stressed Syllable Determination for Romanian Words within Speech Synthesis Applications. *International Journal of Speech Technology,* no 5, pp 237-246.

Ordean, M., Saupe, A., Ordean, M., Duma, M. and Silaghi, Gh. (2009). Enhanced Rule-Based Phonetic Transcription for the Romanian Language. *In Proc. of the 11th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing,* Timisoara, Romania.

Stan, A., Yamagishi, J., King, S., Aylett, M. (2011), The Romanian speech synthesis (RSS) corpus: Building a high quality HMM-based speech synthesis system using a high sampling rate, I*n Speech Communication, vol. 53, no. 3, pp. 442-450*.

Stan, A., Dinescu, F., Tiple, C., Meza, S., Orza, B., Chirila, M., Giurgiu, M. (2017), The SWARA Speech Corpus: A Large Parallel Romanian Read Speech Dataset, *In Proceedings of the 9th Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, Bucharest, Romania.

Toma, S.-A., Oancea, E. and Munteanu, D.-P. (2009). Automatic rule-based syllabication for Romanian. *In Proc of the 5th Conference on Speech Technology and Human-Computer Dialogue,* Constanta, Romania

Toma, S.T., Bârsan, T., Totir, F., Oancea, E. (2013). On letter to sound conversion for Romanian: A comparison of five algorithms. *In Proc. of the 7th Conference on Speech Technology and Human-Computer Dialogue,* Cluj-Napoca, Romania.

Trandabăț, D., Irimia, E., Barbu Mititelu, V., Cristea, D. and Tufiș, D. (2012). The Romanian Language in the Digital Era. *Springer, Metanet White Paper Series.*

Ungurean, C. and Burileanu, D. (2011). An Advanced NLP Framework for High-Quality Text-to-Speech Synthesis. *In Proc. of the 6th Conference on Speech Technology and Human-Computer Dialogue,* Brasov, Romania.

Zhang, R., Li, W., Mo, T. (2018). Review of Deep Learning. arXiv 1804.01653.