https://doi.org/10.17234/INFUTURE.2019.16

Original scientific paper

Automated Phonetic Transcription of Croatian Folklore Genres Using Supervised Machine Learning

Nikola Bakarić
University of Applied Sciences, Velika Gorica, Croatia
nbakaric@gmail.com

Davor Nikolić
Faculty of Humanities and Social Sciences, University of Zagreb, Croatia
dnikoli@ffzg.hr

Summary
This paper aims to detect the possibilities of automatic text transcription for the purpose of preparing a corpus for further natural language processing analysis. The corpus contains various Croatian folklore genres. The transcription goal is to have one character represent one phoneme and remove spaces between accentuated and non-accentuated words. This knowledge independent system is trained using supervised learning methods and applied to the rest of the corpus using classifiers such as the naïve Bayes, k-nearest neighbour, support vector machine and others. The results are compared to a human-annotated sample to determine accuracy.

Key words: text transcription, automation, natural language processing, supervised learning, Croatian folklore genres

Introduction

This paper is a part of a larger research effort which deals with automated classification of Croatian oral literature. In order to approach the problem, the examined corpus of oral literature needed to be prepared, normalised and transcribed to a certain degree. One of the preparatory steps is the syllabification of the corpus of Croatian oral literature. The transcription is an important condition for correct syllabification with regards to pronunciation. The goal is to have one character represent one phoneme. Although Croatian language spelling is mostly phonetic (Pravopis, 2019), there are instances where pronunciation differs significantly. A good portion of such instances can be solved using simple transcription rules, the simplest being the transcription of digraphs lj, nj and dž to ļ, ń and ǯ respectively. There are phenomena, like the yat reflex, which are not so straightforward ("jat," 2019) from a computational perspective and require a more complex approach. Apart from digraphs and the yat reflex, there is phonetic assimilation which occurs in pronunciation when two phonemes form a new sound when spoken together (Yule, 2002). However, the most numerous differences in pronunciation and spelling is the removal of pauses/spaces between accentuated and non-accentuated words, enclitics and proclitics. This is topic of the research presented in this paper.

The problem

As mentioned before, Croatian spelling is phonetic to a very high degree, however there are situations where assimilation and other pronunciation phenomena occur. One of the most common phenomena is the fusion between non-accentuated words (enclitics and proclitics) and their accentuated counterparts in pronunciation. This phenomenon could be described by a simpler rule-based model using their definitions. Enclitics in Croatian language are non-accentuated, present tense and aorist forms of the verbs biti and htjeti, non-accentuated forms of pronouns and the word li (Enklitika, 2019). Proclitics can be monosyllabic, some disyllabic and trisyllabic propositions, conjunctions and particles (Proklitika, 2019). However, both types are ambiguous and can be mistaken for other word types with different functions in pronunciation and spelling. Therefore, supervised machine learning was selected as a more robust and flexible approach to the problem.

Methodology

In order to conduct this preliminary research, a small corpus was prepared. The corpus consisted of 69 blessings and 75 tongue twisters, altogether 1167 words with 1026 occurrences of the space character. It is a part of a larger corpus of Croatian folklore genres collected in the manuscript archives of the Chair of Croatian oral literature at the Department of Croatian Language and Literature, Faculty of Humanities and Social Sciences at the University of Zagreb. A copy of the corpus was further prepared by an expert human annotator who manually marked the instances of the space character which are omitted in pronunciation. At this point, the two copies differed only in the deleted space characters. Table 1 shows several examples of the original and annotated corpus.

Table 1. Examples of annotation

Original text	Annotated
Više ti Bog dao nego što ima zvjezda na nebu.	Višeti Bog dao negoštoima zvjezda nanebu.
Na štriku se suši škotski šosić.	Naštrikuse suši škotski šosić.
Sobzirom na obzir da je moj obzir obzirniji otvog, tvoj obzir kao obzir ne dolazi u obzir.	Sobzirom naobzir dajemoj obzir obzirniji otvog, tvoj obzir kaoobzir nedolazi uobzir.
Prst u pitu, prst u tikvu.	Prst upitu, prst utikvu.
Moja fajfa, stara fajfa, moja fajfa, dobra fajfa. Moja fajfa tak dobre fajfa da ni jedna fajfa ne fajfa tak dobre kak moja fajfa fajfa.	Moja fajfa, stara fajfa, moja fajfa, dobra fajfa. Moja fajfa tak dobre fajfa danijedna fajfa nefajfa tak dobre kakmoja fajfa fajfa.

Source: Authors

The feature selection was based on experience and observation, and two groups of two features were selected. As the non-accentuated words are usually shorter in character length than accentuated words, we selected word length as the first feature group. It consisted of two features, length of the word to the immediate left of the space character and length of the word to the immediate right of the space character. The dataset for word length showed a positively skewed normal distribution of length for both left and right words. The left word length variable set consisted of 14 categories with 4.4 average word length. The right word length variable set consisted of 16 categories with 4.8 average word length.

The second group observed the characters in the immediate left and right of the space. The characters were numerically encoded replacing the letter ‘a’ with 0, ‘b’ with 1, ‘c’ with 2 and so on. Left character variable set consisted of 29 categories with more than half (524) occurrences belonging to the vowels ‘a’, ‘e’ and ‘i’, which is to be expected for Croatian language. The right character variable set was comprised of 25 categories. Here more than half (520) occurrences belonged to consonants such as ‘p’, ‘b’, ‘s’, ‘d’ and ‘k’. The character variable set does not seem to follow normal distribution.

The features were assembled into a sequence of lists (vectors) to which a final value was added, a 0 or 1, depending on the existence of the space character in the annotated corpus. All features and the target value were extracted using custom Python scripts and organised as seen in Figure 1 using the Pandas module for Pyhton (McKinney, 2010).

	left_word	right_word	left_char	right_char	output
0	5	2	12	26	0
1	2	2	12	24	0
2	2	10	8	21	1
3	8	4	11	27	1
4	4	1	0	27	1
...	...	...	...	...	...
1017	5	3	0	21	1
1018	3	6	26	21	1
1019	6	2	0	21	1
1020	2	16	20	21	0
1021	16	5	26	21	1

[1022 rows x 5 columns]

Figure 1. Dataset structure, source: Authors.

The data containing feature vectors and target values was processed using several classification algorithms using the scikit-learn module for Python (Pedregosa et al., 2011). The classification problem was binary as the algorithms had to place each instance of the space character into one of two groups, either deleted or not deleted. The classification algorithms used were:

- Naive Bayes (Gaussian/normal, Multinomial and Complement)
- Support vector machines (Support vector classifier)
- K-nearest neighbour (Nearest centroid classifier)
- Neural network (Multi-layer Perceptron)

The naïve Bayes classifier is one of the simplest, yet most effective classifiers in machine learning tasks (Zhang, 2004), especially in natural language processing. Main feature of this classifier is that it ignores any conditional dependence between observed features thus making it simple yet robust. It has proved to be very successful in many machine learning applications. Several variations of the classifier were tested as its performance depends on the distribution of the input variables.

The scikit-learn tutorial ("Support Vector Machines," 2019) describes the Support vector machines as a set of supervised learning methods used in classification and regression. The support vector classifier module was used. In short, it maps the feature vectors into a model and then finds the margin between two classes. In our case, it used the training set to create a model and calculate a margin. It then mapped the test set vectors to either side of the margin, thus classifying it to the delete space or do-not-delete space group.

The k-nearest neighbour classifier is another simple yet effective method (Goldberger, Roweis, Hinton, Salakhutdinov, 2005) which has several variations. The one used here is the nearest centroid classifier. The method uses the training set to evaluate the nearest neighbours of the test set vector thus determining its class.

Classification using neural networks is slightly more complex than the previous methods. The application of a Multi-layer Perceptron requires additional preparation and fine adjustment of the classifier parameters (LeCun, Bottou, Orr, Müller, 1998). MLP is a non-linear supervised learning algorithm described as a deep neural network with several (at least 3) layers. It learns a function using the training dataset and the provided dimension parameters which passes values along the network nodes (Scikit-Learn Developers, 2018).

All classifiers were applied alongside k-fold cross validation, a method which prevents overfitting in a supervised machine learning environment by separating the dataset into k sections which are then alternated as the training and test sets (Hastie, Tibshirani, Friedman, 2009). The dataset was separated into 10 sections for cross-validation. Each set was used as a test set while the remaining nine sets were used for training. The results presented in the following chapter are the averages of these 10 classification iterations.

Results

One of the aims of this research was to establish the best features for this particular classification experiment. Therefore, the classification algorithms in combination with three different feature sets were applied. The first set of results, presented in Table 2, includes only two features, length of the word to the immediate left of the space character and length of the word to the immediate right of the space character.

Table 2. World lengths as features

Classifier	Accuracy and standard deviation
Multi-layer Perceptron	0.85 (+/- 0.08)
naïve Bayes (Gaussian)	0.84 (+/- 0.07)
Support vector classifier	0.83 (+/- 0.09)
k-NN (Nearest centroid)	0.78 (+/- 0.15)
naïve Bayes (multinomial)	0.75 (+/- 0.09)
naïve Bayes (complement)	0.64 (+/- 0.15)

Source: Authors.

The second set of results, presented in Table 3, includes only the numerically encoded characters left and right of the space character. A general drop in accuracy when compared to the results to Table 2 could be connected to the fact that the set does not seem to follow normal distribution.

Table 3. Characters as features

Classifier	Accuracy and standard deviation
Support vector classifier	0.75 (+/- 0.00)
naïve Bayes (Gaussian)	0.75 (+/- 0.00)
Multi-layer Perceptron	0.75 (+/- 0.01)
naïve Bayes (multinomial)	0.61 (+/- 0.17)
naïve Bayes (complement)	0.56 (+/- 0.18)
k-NN (Nearest centroid)	0.55 (+/- 0.17)

Source: Authors.

Table 4 presents the third set of results which contains a combination of all four features. Here the results are similar to Table 2 where we observed only word length. It could be argued that characters as features do not contribute to, and in some cases decrease the classification accuracy. This indicates that their information value regarding the phenomenon is lower when compared to word length. However, the feature could be simplified by reducing the number of categories, perhaps only to vowels, consonants and punctuation. Another possibility is to scale and weight the features and try to improve accuracy.

Table 4. Word length and characters as features

Classifier	Accuracy and standard deviation
Multi-layer Perceptron	0.85 (+/- 0.09)
Support vector classifier	0.81 (+/- 0.07)
naïve Bayes (Gaussian)	0.80 (+/- 0.09)
naïve Bayes (multinomial)	0.73 (+/- 0.08)
naïve Bayes (complement)	0.65 (+/- 0.14)
k-NN (Nearest centroid)	0.58 (+/- 0.18)

Source: Authors.

Regarding the classifiers, the results show that some handle more features better than others, while some prefer certain types of feature value distributions. The Multi-layer perceptron neural network classifier has proven the most accurate when observing word lengths (Table 2) and the combination of word length and character quality. It is worth noting that initially the naïve Bayes (Gaussian) was the top scoring classifier in the word length environment until we increased the size of hidden layers in the Multi-layer perceptron from (5, 3) to (6, 4). The preparation of the dataset with regards to feature scaling seems to be very important and this leaves room for improvement with the adjustment of parameters for all tested classifiers

Conclusion

Apart from preparing corpuses for academic investigation, which is the main motive of the authors, this preliminary research effort has shown that there are a lot of interesting topics in text to speech translation. This is especially true for small languages which do not have the vast amounts of available lexical data which is the basis of most TTS systems (Mana, Massimino, Pacchiotti, 2001). While the results of certain classifier models are promising, there is room for improvement in dataset preparation, feature selection and tweaking classifier parameters. In order to design a universal model, the corpus should be increased and expanded to include general language.

However, the models relatively high accuracy while using certain classifiers shows promise. The authors plan to develop it further and use it in preparing oral literature corpuses for further analysis. Perhaps a derivation of it will someday be included into a Croatian text-to-speech system.

References

Enklitika. (2019). Hrvatska enciklopedija, mrežno izdanje. Retrieved from http://www.enciklopedija.hr/Natuknica.aspx?ID=17990

Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R. (2005). Neighbourhood Components Analysis. // Advances in Neural Information Processing Systems 17, 513-520. Retrieved from https://cs.nyu.edu/~roweis/papers/ncanips.pdf

Hastie, T., Tibshirani, R., Friedman, J. (2009). Model Assessment and Selection. // The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, 219-257

Jat. (2019). Hrvatska enciklopedija, mrežno izdanje. http://www.enciklopedija.hr/natuknica.aspx?ID=28821

LeCun, Y., Bottou, L., Orr, G., Müller, K. (1998). Efficient BackProp. // Neural Networks: Tricks of the Trade 1998. https://doi.org/10.1192/bjp.112.483.211-a

Mana, F., Massimino, P., Pacchiotti, A. (2001). Using machine learning techniques for grapheme to phoneme transcription. // EUROSPEECH 2001 - SCANDINAVIA - 7th European Conference on Speech Communication and Technology. https://pdfs.semanticscholar.org/ce0e/7ca7c745c2a65b6f3ac7be5df5a8e72065fe.pdf

McKinney, W. (2010). Data Structures for Statistical Computing in Python. // Proceedings of the 9th Python in Science Conference, 51-56. http://conference.scipy.org/proceedings/scipy2010/mckinney.html

Pedregosa, F., Michel, V., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R.,… Duchesnay, Fré. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 12, 2825-2830. http://scikit-learn.sourceforge.net

Pravopis. (2019). Hrvatska enciklopedija, mrežno izdanje. http://www.enciklopedija.hr/natuknica.aspx?id=50013

Proklitika. (2019). Hrvatska enciklopedija, mrežno izdanje. http://www.enciklopedija.hr/natuknica.aspx?id=50588

Scikit-Learn Developers. (2018). Neural network models (supervised). Retrieved September 10, 2019. https://scikit-learn.org/stable/modules/neural_networks_supervised.html

Support Vector Machines. (2019). https://scikit-learn.org/stable/modules/svm.html (10.9.2019)

Yule, G. (2002). The study of languge. Cambridge: Cambridge University Press.

Zhang, H. (2004). The optimality of naive Bayes. In AAAI-04. https://www.cs.unb.ca/~hzhang/publications/FLAIRS04ZhangH.pdf