Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 112 (2017) 1935–1944

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2017, 6-8 September 2017, Marseille, France

Discrimination of Different Serbian Pronunciations from Shtokavian Dialect

Darko Brodić[a,*], Alessia Amelio[b]

[a]*Technical Faculty in Bor, University of Belgrade, V.J. 12, 19210 Bor, Serbia*
[b]*DIMES, University of Calabria, Via Pietro Bucci Cube 44, 87036 Rende (CS), Italy*

Abstract

This paper proposes a new methodology for discrimination of different pronunciations in the Shtokavian dialect of the Serbian language. At the first, the written language (Unicode text) is converted into codes according to the energy status of each character in the text-line. Such a set of codes is seen as a grayscale image. Then, the local structures of the image are explored by local binary operators. It creates a vector set which differentiates various pronunciations of the Serbian language. The experiment is performed on fifty documents given in Serbian language. A comparison performed between the proposed method and the *n*-gram method shows its clear advantage.

Peer-review under responsibility of KES International

Keywords: Image analysis; Natural language processing; Local binary operator; Classification; Document analysis; Serbian language

1. Introduction

Serbian language is a Slavic (Indo-European) language, which belongs to the South Slavic subgroup. It is the standardized variety of the Serbo-Croatian language mainly used by Serbs in Bosnia, Croatia, and Montenegro. However, each of the Serbo-Croatian language variations is called according to the state or nation which somebody belongs to. In spite of differences, the speakers of Serbo-Croatian variations mutually understand each others. Serbian language is an official language in Serbia and one of the three official languages in Bosnia and Herzegovina. Also, it is recognized as a minority language in Montenegro, Croatia, Macedonia, Romania, Hungary, Slovakia and Czech Republic. It is worth noting that some linguists treat Montenegrin as a subpart of the Serbian language. The Serbian language is based on the most widespread dialect called Shtokavian (based on Šumadija-Vojvodina and Eastern Herzegovinian dialects). Furthermore, it is the basis of the Croatian, Bosnian, and Montenegrin languages.

The modern Serbian language has been established in the first part of the 19th century by Vuk Karadžić-Stefanović[1,2,3] and its orthography reformation. Among introducing a new alphabet based on phonemic principles and with a smaller number of letters (Cyrillic alphabet), he introduced the following: (i) the equalization of national and literary lan-

* Corresponding author. Tel.: +381-30-424-555 (ext. 154); fax: +381-30-421-078.
E-mail address: dbrodic@tfbor.bg.ac.rs, aamelio@dimes.unical.it

1877-0509
Peer-review under responsibility of KES International
10.1016/j.procs.2017.08.047

guage, (ii) suspension of all older forms of Serbian literature, and (iii) introduction of the Shtokavian dialect as the only literary dialect in Serbian language.

As its basis, the Shtokavian dialect is used with two different pronunciations: (i) ijekavian, and (ii) ekavian. These pronunciations obtained their names according to the way of changing the previously used letter *jat* into a new letter combination *ije* or *je* in ijekavian pronunciation and *e* in ekavian pronunciation[4]. Currently, the ijekavian and ekavian pronunciation represents an equal literal pronunciation of the Serbian language. However, ekavian pronunciation is used in Serbia, while the ijekavian pronunciation is used outside of Serbia (Bosna in Herzegovina, Montenegro, Croatia). It is worth noting that the differentiation between these pronunciations is the result in the development of phonological systems of the Serbian language. However, it influenced the writing system of the Serbian language, too.

Nowadays, any native speaker of the Serbian language uses either ekavian or ijekavian pronunciation in speech or written language. Hence, their difference is an obvious matter in the everyday communication in Serbian language. It is completely inadequate and strictly unacceptable in the literal language using both pronunciations in one literary work. However, a curiosity is the world literary masterpiece *The Bridge on the Drina* (in Serbian: Na Drini ćuprija) written by Ivo Andrić, a Nobel prize winner in 1961. In this novel, the writer uses ekavian pronunciation as narrator, while the characters use their origin ijekavian pronunciation.

From all aforementioned, it is easy to understand that differentiation of these two pronunciations in the Shtokavian dialect of Serbian language represents a state-of-the-art problem and a real challenge. In this paper, we propose a novel automatic technique to differentiate these two pronunciations in the written Serbian language documents. This method is based on the elements of the previously proposed method for different scripts[5,6,7,8], different languages[9,10,11,12], evolving languages[13] and different orthography recognition[14]. Basically, the proposed method converts each character into an equivalent code according to its energy profile, i.e. Energy Profile Code. The obtained set of codes is seen as an image. Then, the given local image structure is examined by local binary operators such as: (i) Local Binary Pattern (LBP)[15], (ii) Neighbor Binary Pattern (NBP)[16], and (iii) the newly introduced Adjacent Neighbor Binary Pattern (ANBP). Accordingly, various combinations of LBP, NBP and ANBP are explored. The experiment is performed on 50 documents written in different pronunciations of the Shtokavian dialect in Serbian language. Also, the differentiation is made by well-known Natural Language Processing (NLP) techniques such as bi-gram and tri-gram methods[17]. At the end, discrimination of pronunciation is implemented by two different classification tools: (i) Support Vector Machine, and (ii) Naive Bayes.

The paper is organized as follows. Section 2 introduces the proposed method, including script coding, texture analysis for feature extraction, and classification. Section 3 presents the experiment. Section 4 describes the obtained results of the experiment and discusses them. At the end, Section 5 draws conclusions and outlines future work directions.

1.1. Related works

A limited number of works has been introduced in the literature for dialects classification from text document inputs, because most of them classify the dialects according to the speech. In particular, Ref.[18] presents an approach for arabic dialect classification from social media text using semi-supervised learning. It uses multiple classifiers trained by weakly supervised, strongly supervised and unsupervised data. Also, Ref.[19] proposes a method for classification of linguistic varieties spoken in Ethiopia into dialect areas using lexical distances and agglomerative clustering on text strings. Finally, Ref.[20] introduces a supervised machine learning method for recognizing the dialects in the Kurdish texts. Differently from the traditional approaches, we differentiate not a special dialect but pronunciation inside the same dialect, which can be recognized in written as well as in spoken language. To the very best of our knowledge, we are the first to investigate a similar topic in the literature.

2. The Proposed Algorithm

The proposed algorithm consists of the following few stages: (i) Conversion of the Unicode text (printed text only) into the code taking into account the character energy profile, (ii) Extracting of the feature vector using local binary operators, and (iii) Discrimination of the different pronunciation from the feature vector by classification tools.

Fig. 1 shows an overview of the proposed algorithm.

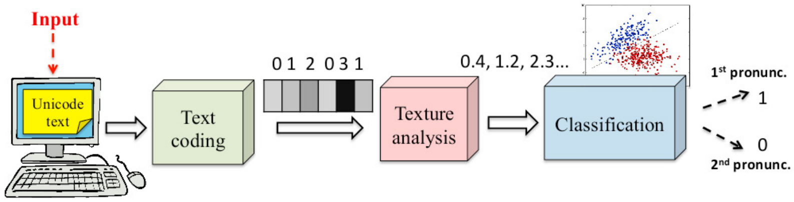

Fig. 1. Overview of the proposed algorithm.

2.1. Conversion of the unicode text

The first stage includes the conversion of each character in the Unicode text into a specific code. This code is based on the position of each character in the text-line. Accordingly, the energy profile for each character based on the typographical characteristics is calculated. Consequently, each text-line can be divided into different zones according to the following virtual lines: (i) top line, (ii) upper line, (iii) base line, and (iv) bottom line. The aforementioned zones are: (i) lower, (ii) middle, and (iii) upper. Then, each letter is classified taking into account which zone(s) it occupies. The following classes of letters exist: (i) base letter in the middle zone, (ii) ascender letter in the middle and upper zones, (iii) descender letter in lower and middle zones, and (iv) full letter in lower, middle and upper zones. They are automatically coded according to Energy Profile Code (EPC) into 0, 1, 2, 3, respectively. It is illustrated in Fig. 2.

Fig. 2. The process of coding letters.

To illustrate the differences between ekavian and ijekavian pronunciations in the Shtokavian dialect of the Serbian language, two text documents are given as well as their equivalent coding according to EPC. This process is shown in Fig. 3.

Statistical analysis of the given text samples is shown in Table 1.

Table 1. Example of the statistical distributions of different letters' types in ekavian and ijekavian pronunciation of Serbian language.

Ekavian	Base letter (0)	Ascender letter (1)	Descender letter (2)	Full letter (3)
No.	39	1	3	2
%	86.67	2.22	6.67	4.44
Ijekavian	Base letter (0)	Ascender letter (1)	Descender letter (2)	Full letter (3)
No.	40	1	3	6
%	80.00	2.00	6.00	12.00

From Table 1, we can notice that the percentage of base letter distribution in ekavian pronunciation is larger approximately between 5-10% compared to ijekavian pronunciation. Furthermore, the percentage of ascender and descendent letter distributions is slightly bigger in ekavian comparing to ijekavian pronunciation. However, the biggest difference can be noticed in the percentage distribution of the full letter. It is almost three times higher in ijekavian pronunciation compared to ekavian one.

In the forthcoming analysis, the coded text has been seen as a grayscale image, where 0, 1, 2, and 3 represent different levels of grays. In this way, we have much wider and more sensitive tools to analyze image texture than

Ово је пример за дечаке и девојчице екавског изговора

100 30 020002 00 000000 0 000030000 00000000 00000020

100300200020000000000000300000000000000000020

(a)

Ово је примјер за дјечаке и дјевојчице ијекавског изговора

100 30 02000302 00 0300000 0 0300030000 030000000 00000020

10030020003020003000000030003000003000000000000000020

(b)

Fig. 3. The pronunciations in the Shtokavian dialect of the Serbian language and their equivalent EPC: (a) ekavian pronunciation, and (b) ijekavian pronunciation.

using usual NLP processing tools like bi-gram, tri-gram, etc. Also, the grayscale images of the two dialects are different, which is the result of different combination and percentage of EPC distributions.

2.2. *Extracting of the feature vector using local binary operators*

The grayscale image is subject to texture analysis for the extraction of the feature vector representing the document. We used different texture operators in this context: (i) co-occurrence matrix [21], (ii) local binary operators [15, 16], and (v) run-length statistics [22, 23, 24]. At the end, we realized that local binary operators obtain the best performances in characterizing the different Serbian pronunciations. In particular, LBP and NBP, together with the new operator ANBP, overcome the performances of the adjacent LBP. Next, we will describe the adopted local binary operators.

2.2.1. *Local binary pattern*

LBP is a texture operator which evaluates the variations of local contrast in the image [15]. It is computed by using a Window of Interest (WOI) of size 3×3 sliding over the image pixel by pixel. Each location of the WOI identifies the center pixel c and compares its intensity with the intensity of the surrounding pixels inside the WOI. The value of each surrounding pixel is set to 1 if its intensity is higher than the intensity of c, otherwise it is set to 0. Then, the values are multiplied by powers of 2 according to their position and summed to generate a binary label for c. Let I_c be the intensity of the center pixel. LBP is computed as follows:

$$LBP_{d,r} = \sum_{i=1}^{d} sign(I_i - I_c) \times 2^{i-1} \tag{1}$$

where I_i is the value of i-th surrounding pixel, d is the number of surrounding pixels, r is the distance between c and the surrounding pixel, and $sign(x)$ is the following function:

$$sign(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \tag{2}$$

Our image is "linear" of size $1 \times N$, where N is the number of the letters in the document, which derives from the aforementioned coding process. Consequently, we only have two surrounding pixels, which are on the left and on the right of c. Hence, in our case $d = 2$ and the number of possible binary labels is $2^2 = 4$. Furthermore, we may consider the surrounding pixels which are adjacent to c ($r = 1$) or more distant from c (r varying from 2 to 5). This allows to explore not only the contrast variations of the closest pixels to c with c itself, but also of the more distant pixels surrounding c with c itself. At the end, the binary label is converted into a decimal label. Then, a histogram counting the frequency of each decimal label inside the image is generated, representing the set of LBP features.

Fig. 4 shows the sample of generation of LBPs when $d = 2$ and $r = 1, ..., 5$.

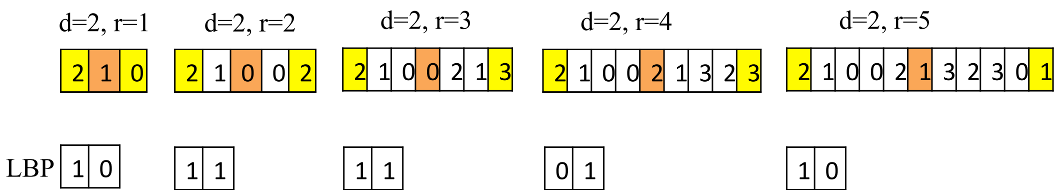

Fig. 4. Sample of generation of LBPs when $d = 2$ and $r = 1, ..., 5$. Surrounding pixels are marked in yellow. The center pixel c is marked in orange.

2.2.2. *Neighbor binary pattern*

NBP is based on the same concepts underlying LBP, with the difference that in NBP a center pixel is not considered for the generation of the binary label. In particular, the local contrast variations between consecutive pixels i and $i + 1$ in the WOI are only considered [16]. Accordingly, the NBP is computed as follows:

$$NBP_{d,r} = \sum_{i=1}^{d} sign(I_i - I_{i+1}) \times 2^{i-1} \tag{3}$$

where I_i is the intensity of i-th pixel, I_{i+1} is the intensity of the consecutive $(i + 1)$-th pixel, and $sign(x)$ is the same function:

$$sign(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \tag{4}$$

Also in this case, because of the "linear" image, we only consider three pixels in the WOI ($d = 2$). Still, it determines a number of possible binary labels which is equal to $2^2 = 4$. Also, we may consider three consecutive adjacent pixels ($r = 1$), or three consecutive pixels which are more distant to each other (r varying from 2 to 5). This takes into account the local contrast variations of the consecutive pixels at different distance. Then, the binary label is converted into a decimal label. Finally, a histogram counting the frequency of each decimal label inside the image corresponds to the set of the NBP features.

Fig. 5 shows the sample of generation of NBPs when $d = 2$ and $r = 1, ..., 5$.

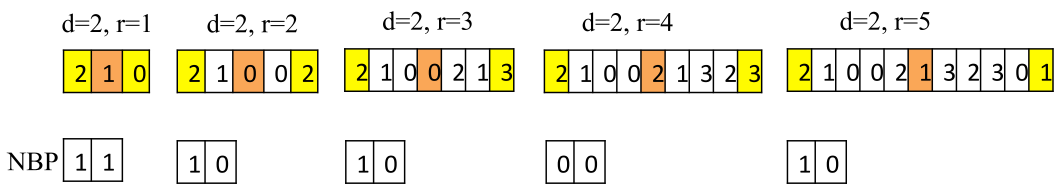

Fig. 5. Sample of generation of NBPs when $d = 2$ and $r = 1, ..., 5$. Consecutive pixels are marked in yellow and orange.

2.2.3. *Adjacent neighbor binary pattern*

Because of the small number of NBP features, we propose a new expansion which is similar to the Adjacent LBP [25]. In particular, we combine two adjacent NBPs in the horizontal direction, i.e. two horizontal consecutive NBPs, in order to create the ANBP, which is a binary label of 4 elements. It extends the number of possible binary labels from $2^2 = 4$ to $2^4 = 16$. Furthermore, we can obtain ANBPs which are composed of NBPs where the consecutive pixels are at different distance ($r = 1, ..., 5$). Still, we only consider three pixels in the WOI which is used for the NBPs ($d = 2$). According to that, a histogram of 16 elements is generated, where the frequency of each corresponding decimal label inside the image is considered. This corresponds to the set of the ANBP features.

Fig. 6 shows the sample of generation of ANBP when $d = 2$ and $r = 1$.

2.3. *Classification*

Classification of the document as given in one of the two Serbian pronunciations is performed by using a binary classification tool. In particular, the feature vector of the document is classified by two example methods: (i) Naive

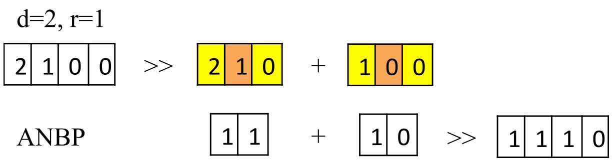

Fig. 6. Sample of generation of ANBP when $d = 2$ and $r = 1$. Consecutive pixels are marked in yellow and orange.

Bayes [26], and (ii) Support Vector Machine [27], which demonstrated their ability and superiority in image and document categorization [28, 29]. Suppose that a set of n training feature vectors with binary class labels in $\{0, 1\}$ is provided, and a new feature vector x_i, composed of k feature values $x_i^1, ..., x_i^k$, is presented for classification as 0 or 1. Next, we will provide the main characteristics of the two classification methods.

2.3.1. Naive bayes

Under the assumption of independence of $x_i^1, ..., x_i^k$ given the class, the Naive Bayes (NB) classifier is as follows [26]:

$$f_{nb}(x_i) = \frac{p(Y = 1)}{p(Y = 0)} \prod_{h=1}^{k} \frac{p(x_i^h|Y = 1)}{p(x_i^h|Y = 0)} \tag{5}$$

where $p(x_i^h|Y = 1)$ is the probability of observing the x_i^h feature value given the class label 1 in the training set, and $p(Y = 1)$ is the probability of observing the class label 1 in the training set. The same is for $Y = 0$.

Accordingly, the feature vector x_i will be classified as 1 if and only if $f_{nb}(x_i) \geq 1$. Otherwise, it will be classified as 0. If the feature values are numerical ones, the probability terms are computed by the normal distribution function:

$$f(w, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(w-\mu)^2}{\sigma^2}} \tag{6}$$

where $p(x_i^h|Y = 1) = f(x_i^h, \mu_{Y=1}, \sigma_{Y=1})$, and $\mu_{Y=1}$ and $\sigma_{Y=1}$ are respectively the mean and standard deviation of the values of h-th feature with class label $Y = 1$ in the training set. The same is for $Y = 0$.

2.3.2. Support vector machine

Let x_j be a feature vector of the training set, with class label y_j. The Support Vector Machine (SVM) finds a solution to the following optimization problem, which is the linear separating hyperplane maximizing the margin [27]:

$$\min_{w,b,\xi} \quad \frac{1}{2} w^T w + C \sum_{j=1}^{n} \xi_j, \quad \text{subject to} \ \ y_j(w^T \phi(x_j) + b) \geq 1 - \xi_j, \quad \xi_j \geq 0 \tag{7}$$

where ϕ is the kernel function mapping the training feature vector x_j into a higher dimensional space, and $C > 0$ is the penalty factor of the training error. Classification of a new feature vector x_i by the SVM is performed using the decision function: $y_i = sgn(w^T \phi(x_i) + b)$, where sgn is the sign function and w, b are the parameters of the learned model.

3. Experiment

The experiment is conducted on a database of 50 documents written in different pronunciations of the Serbian language from Shtokavian dialect [1]. Twenty five out of fifty documents are written in Serbian language with ijekavian pronunciations, while the rest twenty five documents are written in Serbian language with ekavian pronunciations of

[1] The documents from the database are given at the website: `https://sites.google.com/site/documentanalysis2015/serbian-pronunciations-database`

the Shtokavian dialect. Documents are randomly extracted from up-to date daily newspaper articles. The length of the given documents is from 1048 to 4762 characters for ijekavian pronunciations, and from 888 to 7610 characters for ekavian pronunciations. Consequently, typical document length is approximately around 2000 characters[30].

The aim of the experiment is to differentiate documents from the database according to their Serbian pronunciations given in the Shtokavian dialect, i.e. to discriminate documents written in ijekavian (first Serbian pronunciation) from ekavian pronunciation (second Serbian pronunciation).

4. Results and Discussion

The experimentation has been performed in Matlab R2012b on a laptop computer Quad-Core 2.2 GHz, 16 GB RAM and UNIX operating system.

Our aim is the evaluation of the proposed feature representation in characterizing the Serbian pronunciations. Hence, we evaluate the ability of the classifiers in correctly recognizing the Serbian pronunciation of the documents which are represented by our features. Then, we compare our feature representation with the bi-grams and tri-grams normalized frequency vectors, which are well-known models for language identification in the state-of-the-art[31].

Different combinations of feature types (ANBP, NBP, and LBP) with $r = 1, ..., 5$ have been tested for the classification task on the database. At the end, the best combination of the features in terms of performance measures has been selected for each classifier, and definitively adopted in this context. Consequently, each document of the database is represented by a combined vector of ANBP and NBP features with $r = 1$ for the classification by NB method, and by a combined vector of ANBP with $r = 1$, NBP with $r = 1, ..., 3$, and LBP with $r = 3$ for the classification by SVM method.

The feature values have been normalized in the range $[0, 1]$ before the application of any classification task. Each type of feature (ANBP, NBP and LBP) has been separately normalized. In particular, let $\mathcal{X} = \{x_j^p, ..., x_j^q\}$ be a set of features of a given type, corresponding to a subsequence of the feature vector x_j which is the j-th document of the database. Normalization of a given value $x_j^i \in \mathcal{X}$ is performed as follows: $\overline{x_j^i} = \frac{x_j^i}{\sum_{h=p}^{q} x_j^h}$. This value corresponds to the frequency of the i-th pattern inside the j-th document.

Because the feature vectors are composed of numerical values, the NB classifier uses the Eq. (6) for computing the probability terms. The SVM classifier has been tested on the database with different kernel functions, including: (i) linear, (ii) polynomial with order 3, and (iii) Gaussian radial basis with scaling factor of 1. At the end, the linear function demonstrated to obtain the best results in terms of the performance measures. Hence, it has been definitively used in this context.

In order to evaluate the performances of the classification, we generate the confusion matrix between the ground truth classes of the documents and the classification result. From the confusion matrix, we compute three performance measures: (i) precision, (ii) recall, and (iii) f-measure[32]. The precision is the ratio between the relevant documents which are retrieved and the total retrieved documents. The recall is the ratio between the relevant documents which are retrieved and the total relevant documents. The f-measure is the weighted harmonic mean of precision and recall.

In order to make the evaluation independent from the training and test sets, we employ the K-fold cross validation on the database. Hence, the database is randomly divided into K folds. Each fold is used as the test set, while the remaining $K - 1$ folds are merged to create the training set. Each classification algorithm uses the current training set for learning the parameters, and the current test set for computing the performance measures. At the end, we compute the average performance measures on the K runs together with the standard deviation. We set $K = 2, 5, 10$, which are typical values adopted in the literature[33]. Also, we replicate the experiment 50 times for each value of K, for making it independent from the specific division of the database in folds.

Figs. 7 and 8 show the average values of precision, recall and f-measure, together with the standard deviation (see the vertical bars) obtained by the NB algorithm at the different values of K on the database of documents in the first (see Fig. 7) and in the second Serbian pronunciation (see Fig. 8) represented by our feature vectors (ANBP and NBP with $r = 1$), the bi-grams and tri-grams frequency vectors. We can observe that the performance measures do not overcome 0.70 in the case of bi-grams and tri-grams. Also, they decrease as the value of K becomes higher. On the contrary, the performance measures reach up to 0.80 and remain quite stable over the different values of K in the case of our feature representation. Finally, the standard deviation of the performance measures (size of the vertical

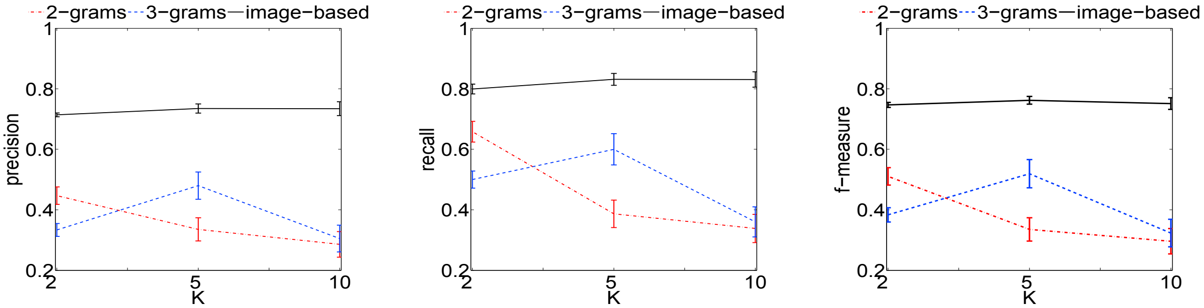

Fig. 7. Classification results obtained by NB algorithm at $K = 2, 5, 10$ of the fold cross validation for the first Serbian pronunciation using our image-based feature representation (ANBP and NBP with $r = 1$), bi-grams and tri-grams in terms of: (a) precision, (b) recall, (c) f-measure. Average results are reported. Vertical bars represent the standard deviation.

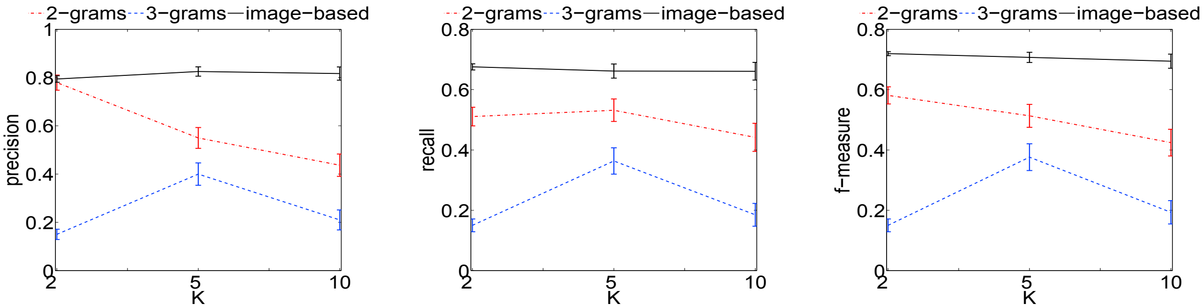

Fig. 8. Classification results obtained by NB algorithm at $K = 2, 5, 10$ of the fold cross validation for the second Serbian pronunciation using our image-based feature representation (ANBP and NBP with $r = 1$), bi-grams and tri-grams in terms of: (a) precision, (b) recall, (c) f-measure. Average results are reported. Vertical bars represent the standard deviation.

bars) is the lowest one for the different values of K in the case of our feature representation (vertical bars are the smallest ones). Hence, our results are more stable than the results obtained by the bi-grams and tri-grams feature representations.

Figs. 9 and 10 show similar results, in terms of average precision, recall and f-measure, together with the standard deviation, obtained by the SVM algorithm at the different values of K on the database of documents in the first (see Fig. 9) and in the second Serbian pronunciation (see Fig. 10) represented by our feature vectors (ANBP with $r = 1$, NBP with $r = 1, 2, 3$, and LBP with $r = 3$), the bi-grams and tri-grams frequency vectors. We can observe that the performance measures never overcome 0.75 for the different values of K in the case of bi-grams and tri-grams. On the contrary, our feature representation obtains values of the performance measures of around 0.70-0.75 for $K = 2$, between 0.75 and 0.80 for $K = 5$, and of 0.80 up to 0.85 for $K = 10$. The values of standard deviation (size of the vertical bars) are quite similar over the different values of K between our feature representation, bi-grams and tri-grams.

If we compare the results obtained by our feature representation using the NB and SVM algorithms, we observe that the NB algorithm has quite stable results over the different values of K of around 0.80 for both the Serbian pronunciations. On the contrary, the results of the SVM algorithm vary from around 0.70-0.75 up to 0.85 for K between 2 and 10. In any case, these results always overcome the results obtained by the competitor methods, demonstrating the efficacy of our feature representation in characterizing the different Serbian pronunciations.

5. Conclusions

This paper proposed a new method for discrimination of different pronunciations in the Shtokavian dialect of the Serbian language. Because it represents a differentiation inside the same language corpora, its solution is a real chal-

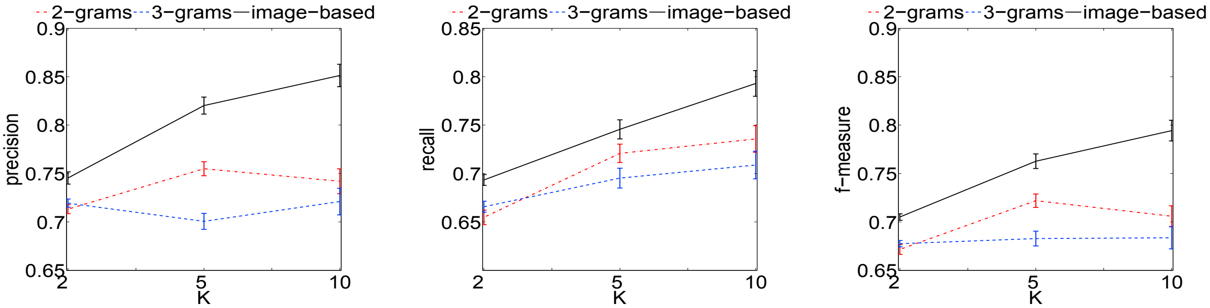

Fig. 9. Classification results obtained by SVM algorithm at $K = 2, 5, 10$ of the fold cross validation for the first Serbian pronunciation using our image-based feature representation (ANBP with $r = 1$, NBP with $r = 1, 2, 3$, and LBP with $r = 3$), bi-grams and tri-grams in terms of: (a) precision, (b) recall, (c) f-measure. Average results are reported. Vertical bars represent the standard deviation.

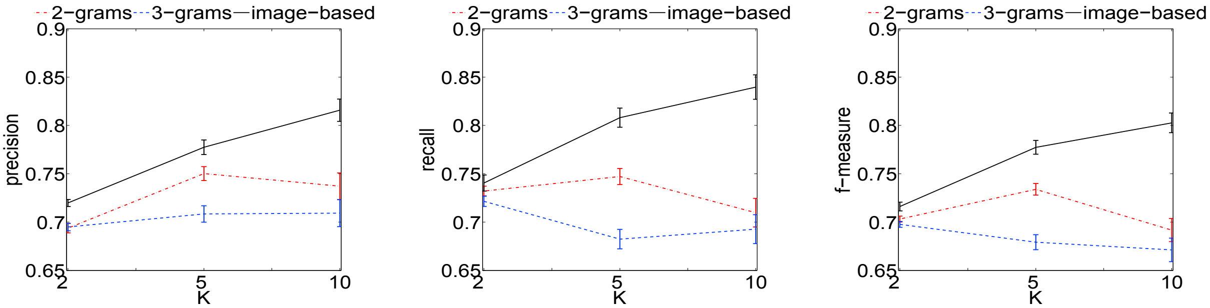

Fig. 10. Classification results obtained by SVM algorithm at $K = 2, 5, 10$ of the fold cross validation for the second Serbian pronunciation using our image-based feature representation (ANBP with $r = 1$, NBP with $r = 1, 2, 3$, and LBP with $r = 3$), bi-grams and tri-grams in terms of: (a) precision, (b) recall, (c) f-measure. Average results are reported. Vertical bars represent the standard deviation.

lenge. The proposed method included the conversion of the letters inside the text into codes representing a grayscale image, the examination of the local image structure by a combination of LBP, NBP and ANBP, and discrimination of the obtained feature vector by Support Vector Machine and Naive Bayes methods. Also, the experiment was conducted on a database of 50 documents given in ekavian and ijekavian pronunciation of the Serbian language. Furthermore, the document database was analyzed with widespread natural language processing tools such as bi-grams and tri-grams. At the end, the results obtained by the proposed method as well as with bi-gram and tri-gram methods were classified by Support Vector Machine and Naive Bayes algorithms. Because the analyzed problem is a real challenge, an f-measure up to 0.85 obtained by the proposed method compared to an f-measure up to 0.75 from bi-gram or tri-gram method represented an obvious progress. Hence, the comparison between the methods showed a clear advantage of the proposed method in all areas compared to bi-gram and trigram methods.

Future work will be oriented to extend the experimentation to other types of dialects, including the dialects from the Southern Italy. Also, the system will be evaluated by other performance measures, such as accuracy and error rate.

Acknowledgements

This work was partially supported by the Grant of the Ministry of Education, Science and Technological development of the Republic Serbia within the project TR33037.

References

1. Stefanović Karadžić, V.. Pismenica serbskoga iezika po govoru prostogă naroda. Wien, 1814. URL: `https://books.google.rs/books?id=PPxDAAAAYAAJ\&pg=PP3\&redir\_esc=y\#v=onepage\&q\&f=false`.

2. Stefanović Karadžić, V.. Srpski rječnik. Wien, 1818. URL: `https://archive.org/details/srpskirjecnik00kara`.
3. Stefanović Karadžić, V.. Prvi srpski bukvar. Beč, 1827. URL: `http://documents.tips/documents/vukov-srpski-bukvar-562f9a484c636.html`.
4. Lj. Subotić D. Sredojević, I.B.. Phonetics and phonology (in serbian: Fonetika i fonologija: ortoepska i ortografska norma standardnog srpskoj jezika). In: *Univerzitet u Novom Sadu, Filozofski fakultet*. 2012, .
5. Brodić, D., Milivojević, Z., Amelio, A.. Analysis of the south slavic scripts by run-length features of the image texture. *Elektronika ir Elektrotechnika* 2015;**21**(4). doi:`10.5755/j01.eee.21.4.12785`.
6. Brodić, D., Milivojević, Z.N., Maluckov, Č.A.. An approach to the script discrimination in the slavic documents. *Soft Computing* 2015; **19**(9):2655–2665. doi:`10.1007/s00500-014-1435-1`.
7. Brodić, D., Amelio, A., Milivojević, Z.N.. Identification of fraktur and latin scripts in german historical documents using image texture analysis. *Applied Artificial Intelligence* 2016;**30**(5):379–395. doi:`10.1080/08839514.2016.1185855`.
8. Brodić, D., Amelio, A., Milivojević, Z., Jevtic, M.. Document image coding and clustering for script discrimination. *ICIC Express Letters* 2016;**10**(7):1561–1566.
9. Brodić, D., Amelio, A., Milivojević, Z.N.. *Characterization and Distinction Between Closely Related South Slavic Languages on the Example of Serbian and Croatian*. Cham: Springer International Publishing. ISBN 978-3-319-23192-1; 2015, p. 654–666. doi:`10.1007/978-3-319-23192-1\_55`.
10. Brodić, D., Amelio, A., Milivojević, Z.. A new image analysis framework for latin and italian language discrimination. vol. 1589. 2016, p. 46–55.
11. Brodić, D., Amelio, A., Milivojević, Z.N.. Language discrimination by texture analysis of the image corresponding to the text. *Neural Computing and Applications* 2016;:1–22doi:`10.1007/s00521-016-2527-x`.
12. Brodić, D., Amelio, A., Milivojević, Z.N.. An approach to the language discrimination in different scripts using adjacent local binary pattern. *Journal of Experimental & Theoretical Artificial Intelligence* 0;**0**(0):1–19. doi:`10.1080/0952813X.2016.1264090`.
13. Brodić, D., Amelio, A., Milivojević, Z.. Clustering documents in evolving languages by image texture analysis. *Applied Intelligence* 2016; :1–18doi:`10.1007/s10489-016-0878-8`.
14. Brodić, D., Amelio, A.. Dating the historical documents from digitalized books by orthography recognition. In: *13th Italian Research Conference on Digital Libraries*. 2017, p. 1–12. URL: `http://ircdl2017.unimore.it/papers/IRCDL_2017_paper_2.pdf`.
15. Ojala, T., Pietikäinen, M., Mäenpää, T.. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans Pattern Anal Mach Intell* 2002;**24**(7):971–987. doi:`10.1109/TPAMI.2002.1017623`.
16. Hamouchene, I., Aouat, S.. A new approach for texture segmentation based on nbp method. *Multimedia Tools Appl* 2017;**76**(2):1921–1940. doi:`10.1007/s11042-015-3185-1`.
17. Roark, B., Saraclar, M., Collins, M.. Discriminative n-gram language modeling. *Comput Speech Lang* 2007;**21**(2):373–392. doi:`10.1016/j.csl.2006.06.006`.
18. Huang, F.. Improved arabic dialect classification with social media data. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics; 2015, p. 2118–2126. URL: `http://aclweb.org/anthology/D15-1254`.
19. Negesse, F.. Classification of oromo dialects: A computational approach. *International Journal of Computational Linguistics (IJCL)* 2015; **6**(1):1–10.
20. Hassani, H., Medjedovic, D.. Automatic kurdish dialects identification. *Computer Science & Information Technology* 2016;:61.
21. Haralick, R.M., Shanmugam, K., Dinstein, I.. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics* 1973;**SMC-3**(6):610–621. doi:`10.1109/TSMC.1973.4309314`.
22. Galloway, M.M.. Texture analysis using gray level run lengths. *Computer Graphics and Image Processing* 1975;**4**(2):172 – 179. doi:`10.1016/S0146-664X(75)80008-6`.
23. Chu, A., Sehgal, C., Greenleaf, J.. Use of gray value distribution of run lengths for texture analysis. *Pattern Recognition Letters* 1990; **11**(6):415 – 419. doi:`10.1016/0167-8655(90)90112-F`.
24. Dasarathy, B.V., Holder, E.B.. Image characterizations based on joint gray level-run length distributions. *Pattern Recognition Letters* 1991; **12**(8):497 – 502. doi:`10.1016/0167-8655(91)80014-2`.
25. Nosaka, R., Ohkawa, Y., Fukui, K.. *Feature Extraction Based on Co-occurrence of Adjacent Local Binary Patterns*. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-25346-1; 2012, p. 82–91. doi:`10.1007/978-3-642-25346-1\_8`.
26. Russell, S.J., Norvig, P.. *Artificial Intelligence: A Modern Approach*. Pearson Education; 2 ed.; 2003. ISBN 0137903952.
27. Cortes, C., Vapnik, V.. Support-vector networks. *Machine Learning* 1995;**20**(3):273–297. doi:`10.1023/A:1022627411411`.
28. Khan, A., Baharudin, B., Lee, L.H., Khan, K., Tronoh, U.T.P.. A review of machine learning algorithms for text-documents classification. In: *Journal of Advances In Information Technology, VOL*. 2010, .
29. Kim, J., Kim, B.S., Savarese, S.. Comparing image classification methods: K-nearest-neighbor and support-vector-machines. In: *Proceedings of the 6th WSEAS International Conference on Computer Engineering and Applications, and Proceedings of the 2012 American Conference on Applied Mathematics*; AMERICAN-MATH'12/CEA'12. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS). ISBN 978-1-61804-064-0; 2012, p. 133–138.
30. Document analysis website. 2015. URL: `https://sites.google.com/site/documentanalysis2015/home`.
31. Turney, P.D., Pantel, P.. From frequency to meaning: Vector space models of semantics. *J Artif Int Res* 2010;**37**(1):141–188.
32. Sokolova, M., Lapalme, G.. A systematic analysis of performance measures for classification tasks. *Information Processing & Management* 2009;**45**(4):427 – 437. doi:`10.1016/j.ipm.2009.03.002`.
33. Yadav, S., Shukla, S.. Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. In: *2016 IEEE 6th International Conference on Advanced Computing (IACC)*. 2016, p. 78–83. doi:`10.1109/IACC.2016.25`.