Int J Speech Technol (2016) 19:687–695
DOI 10.1007/s10772-016-9360-6

Maghrebian dialect recognition based on support vector machines and neural network classifiers

Mohamed Hassine[1] · Lotfi Boussaid[2] · Hassani Messaoud[1]

Received: 18 March 2016 / Accepted: 8 August 2016 / Published online: 16 August 2016

Abstract This paper investigates the feed forward back propagation neural network (FFBPNN) and the support vector machine (SVM) for the classification of two Maghrebian dialects: Tunisian and Moroccan. The dialect used by the Moroccan speakers is called "La Darijja" and that of Tunisians is called "Darija". An Automatic Speech Recognition System is implemented in order to identify ten Arabic digits (from zero to nine). The implementation of our present system consists of two phases: The features extraction using a variety of popular hybrid techniques and the classification phase using separately the FFBPNN and the SVM. The experimental results showed that the recognition rates with both approaches have reached 98.3 % with FFBPNN and 97.5 % with SVM.

Keywords Maghrebian dialect recognition · Feed forward back-propagation neural network (FFBPNN) · Support vector machines (SVM) · Principal component analysis (PCA)

✉ Lotfi Boussaid
lotfi.boussaid@enim.rnu.tn

Mohamed Hassine
hassinemohamed60@yahoo.com

Hassani Messaoud
hassani.messaoud@enim.rnu.tn

[1] Laboratory of Automatic Signal and Image Processing, National School of Engineers of Monastir, University of Monastir, 5019 Monastir, Tunisia

[2] Laboratory of Electronics and Microelectronics, Faculty of Science of Monastir, University of Monastir, 5019 Monastir, Tunisia

1 Introduction

Arabic is the sixth most widely spoken language in the world. It is the language of over 24 countries and spoken by more than 3000 millions of speakers. In Arabic language, we distinguish three varieties. The Classical Arabic (CA), which is the language of the Curran, the Modern Standard Arabic (MSA), usually used in media and studied at school, and the dialect.

There are two major groups in Arabic dialect (AD): the western AD (Maghreb or North Africa) and the eastern one (Levantine Arabic, Gulf Arabic and Egyptian Arabic). In this study, we focus on the dialect of two Maghrebian countries: Tunisia and Morocco. The dialect is the daily spoken language which is different from CA and modern Arabic. The principal Tunisian AD called "Darija" and sometimes "Tounsi" has two important forms, which are the urban dialect and the rural dialect. There are also some regional variations in these dialects: The variety of Tunis region, the Sahelian variety, the Sfaxian, etc.

The Moroccan Arabic is characterized by the competition of Arabic, Berber and French. Moroccan dialect consists especially of Darija AD (Cobert 2003) and Tamazight (Berber dialect) (Amour et al. 2004; Boukous 1998) distinguished five main dialect varieties in Morocco: The urban speaking called (Mdini), the mountain speaking called (Jebli), the bedouin talks called (arubi, bedwi), the Hassane variety that is called (aribi). The diversity of dialects in Maghreb is related to the presence of the French language, which is the language of the old colonial power and the existence of Substratum Berber.

The aforementioned dialects are significantly different from MSA on all linguistic levels and from the point of view of vocal or transcription forms. We distinguish the absence of the contingencies endings, modification of the paradigm of the conjugation, different order of words in the sentence, the use of terms borrowed from western languages (Hamdi 2007). Moreover, several informal transcription forms of AD are today observed. Users of AD online write in different scripts (Arabic, Romanizations interspersed with digits), they also sometimes write phonemically. Similar to other languages (not unique to AD) in these informal genres, we observe rampant speech effects such as elongations and the use of emoticons within the text which compounds the problem further for processing AD (Dasigi and Diab 2011). Our paper is motivated by the crucial complexity of AD and presents an initial attempt at classifying Moroccan and Tunisian Dialects.

This paper is organized as follows: In Sect. 2, we shed some lights on basic related works. In Sect. 3, we describe some theoretical backgrounds. The suggested methodology is detailed in Sect. 4, while Sect. 5 describes experimental results. The main results are discussed in Sect. 5. Finally, conclusion is given in Sect. 6.

2 Related works

Most studies and processing tools for Arabic language classification are designed for MSA and a little researches addressed AD classification because it is an under resourced language (Lack of experimental corpora for Maghrebian dialect). What follows is a literature review of previous works examining ADs identification and classification. Previously (Barkat-Defradas et al. 2004) were interested in identifying two ADs: Middle-east Arabic and Maghrebian Arabic based on phonetic, phonologic and rhythmic features. Their work was carried out in two methods: perceptual identification and automatic identification. During the perceptual experience the two dialects were distinguished in two geographic zones with 98 % performance. The automatic identification was performed by the automatic detection of vowel segments, the Frequency cepstral coefficients (MFCC and ΔMFCC), the derivative of energy, the duration of vowel segments and finally by a pseudo-syllables representation. In this case, the system realized a classification rate by vowel and by rhythm parameters of 82 and 73 % respectively. It concluded that the distribution of the vowels in the acoustical space, the ratio of long vowels by short vowels and the rhythm were of primordial importance in dialect discrimination.

Alorifis and et al. utilized a combination of the MFCC, ΔMFCC, energy and an ergodic HMM to build an AD identification system focusing on two major Dialects: Egyptian dialect and Gulf dialect (Alorifis 2008). The identification system reached a result of 96.67 % correct identification. In another research developed by (Sadat et al. 2014), it is a comparative study to identify ADs of eighteen Arabic countries based on social media texts. Two methods were used: The character n-gram Markov language models and the naïve Bayes classifier using three n-gram models, uni-gram, bi-gram and tri-gram. Firstly it was shown that naïve Bayes classifier performs better than n-gram Markov models, secondly with naïve Bayes classifier based on character bi-gram model, more accurate results were obtained than with other classifiers (uni-gram, bi-gram) and provides an F-measure of 80 % and an accuracy of 98 %.

Biadsy and et al. showed in their paper that four Arabic colloquial ADs (Gulf,Iraqi, Levantine and Egyptian) and MSA can be distinguished using a phonotactic approach with a good accuracy (Hirschberg and Habash 2009). They employed a parallel phone recognizer (PRLM), and found that the most distinguishable dialect among the five variants considered here is MSA (F-measure is always above 98 %). Egyptian Arabic is second (F-measure 90.2 %) followed by Levantine (F-measure of 79.4 %).

Habash and et al. presented CODA, a Conventional Orthography for writing AD (Habash et al. 2009). Pasha and et al. presented "MADAMIRA" (Pasha et al. 2014), a system for morphological analysis and disambiguation of AD that combine the best aspects of two systems. The first is "MADA" (Habash et al. 2013), which adopts a morphological analyzer for MSA written texts and apply SVM and N-gram models to produce a prediction per word for different morphological features. The second is "AMIRA which is a written text analyzer that takes mainly a multistep approach to tokenization, part-of-speech tagging and lemmatization (Diab et al. 2007).

A supervised system with a Naïve Bayes classifier was implemented for performing identification between MSA and Egyptian Arabic dialect (EAD) based on sentences level. This system uses two kinds of features: Core Features and Meta Features to train the classifier to predict the sentences labels of an input text. The obtained results reached 85.5 % of correct identification (Elfardy and Diab 2103).

Dasigi and et al. built a system, which is called CADACT to identify orthographic variations in AD texts by clustering the similar strings (Dasigi and Diab 2011). Three basic similarity measures were used: String based Similarity as direct Levenshtein Edit Distance; string based similarity biased edit distance; and contextual string similarity. To measure the performance of this system, two ADs were targeted: Egyptian Arabic dialect (EAD) and Levantine Dialect. The system achieved the highest Entropy of 0.19 for Egyptian (corresponding to 68 % cluster precision) and Levantine (corresponding to 64 % cluster precision) respectively.

H. Bŏril and al investigated the Arabic language identification (LID) and AD identification (DID) using two corpora (Boril et al. 2012). The first one contains Conversational Telephone Speech (CTS) collections of ADs distributed through the Linguistic Data Consortium (LDC) and the second was an in house pan-arabic corpus. Their work is conducted in two parts: The first was concerned by analyzing channel and noise characteristics of the already mentioned corpora. It was found firstly that these characteristics are unique and fairly distinctive for each dialect corpus, secondly silence regions were found to be sufficient information for DID with 83 % average accuracy. In the second part of the study, a phonotactic recognition system is introduced where 9 non-Arabic phone recognizers and 4 SVM classifiers were tested on the two aforementioned corpora. In this case, the system realized 14.5 and 32.3 % average error rates for the LDC's and Pan-Arabic corpora respectively (Boril et al. 2012).

Lachachi and et al. presented two approaches based on SVM multi-class for reducing the Universal Background Model in the automatic dialect identification system (Nour-Eddine and Abdelkader 2015). The approaches were tested on five Maghrebian dialects of spontaneous conversation and the results are compared to those of the base line system. The results proved that both approaches realized an improvement of dialect identification, an absolute precision of 80.49 % for the first and 74.99 % for the second.

3 Theoretical background

3.1 Features extraction phase

To extract the relevant information, minimize noise and remove redundancy from the speech, feature extraction methods are needed. These methods can be used separately or jointly such as: The perceptual linear prediction and its first order temporal derivative (PLP and ΔPLP), the relative spectral perceptual linear prediction (Rasta-PLP), the Mel frequency cepstral coefficients and its first order temporal derivative (MFCC and ΔMFCC). Besides, to reduce dimensionality by preserving the main intrinsic information in the speech signal, the obtained features were followed separately by vector quantization of Linde-Bruze and Gray (VQLBG) and by principal component analysis (PCA).

3.1.1 The mel frequency cepstral coefficients (MFCCs)

The MFCCs are dominant in speech recognition areas; this feature extraction technique uses a non-linear frequency scale which is the Mel scale in order to simulate the frequency response of the human auditory system. MFCCs are based on known variation of the human ear's critical bandwidth with frequency (Price and Sophomore student 2005; Lindasalwa et al. 2010). It is a psychoacoustic measure of pitches judged by human that is linear in bottom of 1000 Hz (Semet and Treffo 2002) and logarithmic above. The MFCCs provide a compact representation of the given speech signal.

We describe as follows the different steps used to extract the MFCCs features from the ten Arabic digits from zero to nine.

3.1.1.1 Pre-emphasis Each signal corresponding to each digit is pre-emphasized to increase the contribution of the high frequencies in the speech signal.

3.1.1.2 Windowing In this stage, the pre-emphasized signal is divided into frames of 25 ms and multiplied by an overlapped sliding Hamming window with an overlapping step of 10 ms to avoid leakage and spectral distortion at the beginning and at the end of each frame. The Hamming window is given by:

$$h(n) = \begin{cases} 0.54 - 0.46\cos\pi\left(\dfrac{2\pi n}{N-1}\right) & \mathit{if}\ 0 \le n \le N-1 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where N is the number of samples in the window.

This window was chosen since it generates lesser oscillations than other windows and has reasonable side lobe and main lobe characteristics which are required for the DFT computation. The hamming window has effectively better selectivity for large signals and is commonly used is speech processing (Hassine et al. 2015).

3.1.1.3 Discrete Fourier transform (DFT) DFT is used to convert each frame of N samples from the time domain to the frequency domain, which yields to the signal spectrum.

3.1.1.4 Mel filters bank Since the frequencies range obtained in the previous step is wide, a filter bank in the Mel scale is built to pass the speech signal through it and to avoid huge calculations. The Mel filters bank are series of overlapped triangular filters which are built in such a way that the low boundary of a filter is situated at the center of the previous filter and the upper boundary is at the next filter.

3.1.1.5 Discrete Cosine transform (DCT) In this step the discrete Cosine transform is done which yields to the Mel Cepstral coefficients.

3.1.1.6 The first-order temporal derivative coefficients of MFCCs (ΔMFCCs) ΔMFCCs are also known as differential coefficients. They correspond to the trajectories of the basic MFCCs coefficients over the time (Srinivasan 2011).

To calculate the delta coefficients, the following formula is used:

$$d_i = \frac{\sum_{n=1}^{N} n(C_{n+i} - C_{n-i})}{2\sum_{n=1}^{N} n^2} \quad (2)$$

where d_i is the delta coefficient at frame *i* computed in terms of the corresponding basic Cepstral Coefficients c_{n+I} to c_{n-i}. A typical value for N is 2.

3.1.2 Perceptual Linear prediction coefficients (PLP)

Perceptual Linear prediction (PLP) coefficient is another feature extraction technique which emulates the human auditory system and uses a Bark scale which is different from the Mel scale used in MFCCs. There are three main concepts behind PLP (Gunawan and Hasegawa-Johnson 2001). They are critical band frequency

The following steps describe the computing of PLP coefficients:

The three first steps are similar to that of the MFCCs, the difference here is the use of a filter bank in Bark scale instead of Mel scale and the remaining steps are:

3.1.2.1 Equal-loudness curve The role of equal-loudness curve is to approximate the sensitivity of human hearing at various different frequencies.

3.1.2.2 Intensity-loudness power law In this step the non-linear relationship between signal intensity and perceived loudness is applied (Hermansky 1990).

3.1.2.3 Inverse discrete Fourier transform (IDFT) and Cepstral analysis Here, an inverse discrete Fourier transform (IDFT) is applied to the result found in the previous stage and then Levinson-Durbin Algorithm is applied to the obtained result to compute the linear prediction coefficients (LPC). These will be converted to PLP cepstral coefficients (Antoniol et al. 2005).

3.1.2.4 The first-order temporal derivative coefficients of PLPs (ΔPLPs) The PLP feature vector describes only the power spectral envelope of a single frame, however differential coefficients give information about the dynamic evolution of the PLP coefficients over the time. ΔPLPs coefficients are computed similarly as in Eq. 1.

3.1.3 Principal component analysis (PCA)

The principal component analysis technique is used for reducing dimensionality of the obtained features by conserving the intrinsic original information. We used PCA as a modeling tool of the extracted features because it is a simple, non-parametric method of extracting relevant information from confusing data sets. Here our purpose of using PCA is to facilitate the recognition since this technique allows us to represent each letter by a minimum number of vectors (Shlens 2003).

In practice and to apply PCA, we follow the steps below:

- Calculate the covariance matrix of the features on which we will apply PCA;
- Find the eigenvectors of the obtained covariance matrix;
- Extract diagonal of matrix as vector;
- Sort the variances in decreasing order;
- Project the original data set.

3.1.4 Vector quantization (VQ)

Vector quantization is a process of mapping vectors from a vector space to a finite number of regions in that space. Here the LBG algorithm is used and implemented by the following recursive procedures:

1. Design a 1-vector codebook: this is the centroid of the entire set of training vectors (hence, no iteration is required here).
2. Double the size of the codebook by splitting each current codebook Y_n according to the rule:

   $$Y_n^+ = Yn(1+\varepsilon)$$
   $$Y_n^- = Yn(1-\varepsilon)$$

   Where *n* varies from 1 to the current size of the codebook, and ε is a splitting parameter (we choose ε = 0.01).
3. Nearest-Neighbor Search: for each training vector, find the codeword in the current codebook that is closest (in terms of similarity measurement), and assign that vector to the corresponding cell (associated with the closest codeword).
4. Centroid Update: update the codeword in each cell using the centroid of the training vectors assigned to that cell.
5. Repeat steps 3 and 4 until the average distance falls below a preset threshold
6. Repeat steps 2, 3 and 4 until a codebook size of *M* is designed (Ameen et al. 2012).

3.2 Recognition phase

In the recognition phase, we have used two methods based on feed-forward back propagation neural networks (FFBPNN) and the support vector machine (SVM).

3.2.1 Neural network

The FFBPNN is the most popular Multilayer architecture used for automatic speech recognition. It is formed by an input layer (Xi), one intermediary or hidden layer (HL) and an output layer (Y). A weight matrix (W) can be defined for each of these layers.

This artificial neural network topology can solve classification problems involving non-linearly separable patterns and can be used as a universal function generator (Haykin 2009). Important issues in MLP design include specification of the number of hidden layers and the number of units in these layers. The number of input and output units is defined by the problem (there may be some uncertainty about precisely which inputs to use) (Venkateswarlu et al. 2011).

3.2.2 The support vector machine (SVM)

The support vector machine is a statistical method for supervised binary classification (Yu and Kim 2012) based on two notions: The margin maximization and the kernel functions. The margin is the distance between the separation border of the two classes and the nearest samples called the support vector machines. When the problem is linearly separable the SVM algorithm searches the optimum hyperplan which separates the two classes, while in non linearly ones it uses a kernel function to project data in a high dimension space where they become linearly separable. The SVM is used in numerous learning problems, particularly in speech recognition, face recognition, bioinformatics and medical diagnostics (Burges 1998; Smola and Scholkopf 1998).

Mathematically the SVM consists of building a function f which corresponds to an output

y $= \{-1,\ 1\}$ for each input $X \in \Re^d$

$$f : \Re^d \rightarrow \{-1, 1\}$$
$$X \rightarrow y$$

So a discrimination function h is obtained by linearly combination of an input vector $X = (x_1, x_2, \ldots xd)$.

$$h(X) = wX + b \tag{3}$$

Where w is a vector perpendicular to the hyperplan,b is a bias vector and $h(X)$ is the hyperplan equation.

The class of X is determined by the sign of $h(X)$, if $h(X) \geq 0$ then $X \in$ class $+1$ otherwise.

$X \in$ class $- 1$.

4 Methodology

The methodology of our work is performed in the following steps:

4.1 Speech recording and preprocessing

For the speech corpus, we choose the ten digits from zero to nine. The recording was realized in suitable conditions where professional acoustical materials were used. A digital mixing console (Studer on air 2000 M2), a dynamic microphone (MD 421) and a professional software (Sound Forge 6.0). The speech was recorded in Mono wave files, at a sampling rate of 44100 Hz and coded in 16 bits.

Since AD is under-resourced language, we prepared a proper corpus: where eight speakers: four Tunisians (2 males and 2 females) and four Moroccans (2 males and 2 females) pronounced the ten digits five times in their corresponding dialects (Tunisian or Moroccan dialect). The speech signal of each digit was stored in a proper wave file. A trial consists in pronouncing all the digits by one speaker one time, so each speaker participates in the present corpus by five trials, hence the digits corpus involves 40 trials which counts 400 wave files. The training corpus was built by 80 % of the entire digits corpus. The test corpus consisted of 20 % of the entire digits corpus which have not been in the training corpus. The validation corpus represents 20 % of the entire digits corpus and it involves data which have been in the training corpus.

Table 1 describes the SAMPA phonetic transcription of the ten digits in Tunisian and Moroccan dialects.

4.2 Applying the features extraction techniques

After pre-emphasizing the speech signal of each digit, the already mentioned feature extraction techniques were applied using two hybrid techniques: Mel Frequency Cepstral Coefficients or Perceptual Linear Prediction followed by their first temporal derivative (MFCC + ΔMFCC) and (PLP + ΔPLP). In this stage a matrix of features including 26 lines and a variant column number is obtained for each digit speech signal. In order to reduce dimensionality of these features and redundancy by

Table 1 SAMPA phonetic transcription of the digits in both dialects

Digits	Tunisia	Morocco
0	sfir	sifr
1	wa:Xid	waXd
2	Tni:n	zuz
3	tla:Ta	tlata
4	?arb?'a	Rb?'a
5	xamsa	xamsa:
6	Sitta	Stta
7	Sab?'a	Sb?'a
8	Tmanja	tmnja
9	Tis?'a	Ts?'ud

conserving the intrinsic information, the Vector Quantization algorithm of Linde-Bruze and Gray (LBG) and the principal component analysis technique (PCA) were used separately. Finally, the obtained vectors were concatenated in order to represent each digit with one vector only.

4.3 Applying FFBPNN

The features already found (in the feature extraction section) were stored in a matrix which is then composed of vectors corresponding to all the digits (one vector for each digit). The latter matrix was provided to the neural network as an input. In our work, FFBPNN has been trained in supervised mode. We used the binary code of 7 bits: (1,000,000) as a Target for all the vectors that represent the Tunisian dialect and a code of (0,100,000) for that of the Morrocan dialect. We chose a number of neurons between 70 and 90 and the Tangent sigmoid "TanSig" activation function for the hidden layer. For the output layer, we chose seven neurons and the logistic sigmoid "LogSig" activation function. The learning algorithm was stochastic gradient descent and the used epochs have been varied between 23 and 41. The performance function is mean square error (MSE) and the training function is that of Levenberg–Marquardt 'Trainlm'. The remaining parameters are taken by default.

A MATLAB Neural Network Toolkit has been used. The feed-forward N-layer NN is created by newff MATLAB command (2006).

4.4 Applying SVM

In this stage, OSU-SVM toolbox for matlab is applied. It uses the algorithms of the package LIBSVM of Chih-Jen Lin and Chih-Chung Chang (Chang and Lin 2004). These routines are coded in C++ and OSU-SVM toolbox is used through Malab MEX libraries, which allows for greater speed calculation and better memory management.

OSU-SVM includes some Matlab functions which are easy to use, such as functions for learning. In this section, the radial basic function kernel RBFSVC is used. This function described by the following equation:

$$RBFSVC(U,V) = e^{-\gamma|U-V|^2} \qquad (4)$$

U, V are two vectors and γ is a constant.

RBFSVC takes as inputs, the digits features, the corresponding labels,γ and the parameter C which controls the "trade-off" between classification error and the security margin.

OSU-SVM includes some other functions which are used in this work respectively after RBFSVC.

- *The test function* SVMTest, which provides the classification rate and the confusion matrix for every classification process.
- *The classification function* SVMClass, which takes the digits features and some parameters as inputs and provides a vector of labels corresponding to the classification result.

The features used in the training phase consist of 160 vectors corresponding to the Tunisian dialect and labeled by "1" and 160 other vectors to the Moroccan dialect labeled by "−1", meanwhile the features in the test phase consists only of 40 vectors for each dialect labeled by "1" and "−1" respectively for the Tunisian and the Moroccan dialect.

5 Results and discussion

Our work is conducted in two experiments where features extraction and classification were implemented in Matlab Software 7. In the first experiment, the classification was realized using FFBPNN. We let Matlab program prepared for our classification system running until one of the known multi-layer perceptron (MLP) stop criteria is reached. We notice then each time the corresponding error rates.

The stop criteria imply training stops early when any of these conditions occurs:

- The maximum number of epochs (repetitions) is reached.
- The maximum amount of time for training the FFBPNN is exceeded.
- The Performance is minimized to the goal.

Table 2 Classification performance with FFBPNN

Techniques	Training error (%)	Validation error (%)	Test perf. (%)	Rec. time (s)	Epochs
PLP + ΔPLP + PCA + FFBPNN	139.223e−016	214.80−017	98.18	7159.920543	38
MFCC + ΔMFCC + PCA + FFBPNN	753.987e−015	815.53e−015	96.94	7287.829331	41
PLP + ΔPLP + VQLBG + FBPNN	264.782e−014	1.1669e−014	**98.30**	3628.540361	23
MFCC + ΔMFCC + VQLBG + FFBPNN	0.00446429	0.0018	97.07	5000.895744	31

Bold value indicates the best performance obtained by using these techniques

- The performance gradient falls below minimal gradient (min_grad).
- "mu" exceeds mu max, where mu is the learning rate.
- The validation performance has increased more than max_fail times since the last time it decreased (when using validation).

The parameters maximum number of epochs, maximum amount of time, goal, min_grad, mu max and max_fail time are chosen by the user.

During the whole experiment, we noticed that PLP followed firstly by Δ PLP and secondly by VQLBG has realized the best classification rate which is 98.30 %. PLP followed firstly by ΔPLP and secondly by PCA has occupied the second order in term of performances with 98.18 % classification rate. the stop criterion was "reaching min_grad".

In Table 2, the training error, the validation error, the test performance (classification success) and the computational time (Rec.time) taken by the FFBPNN are presented with each hybrid technique.

In Fig. 1, the different lines blue, green and red correspond respectively to training error, validation error and test error.

In the second experiment, the automatic classification was realized by SVM. The best classification rate has been obtained by PLP followed firstly by Δ PLP and secondly by PCA with 97.5 %. In this experiment, PCA proved to be better than VQLBG which realized 81.25 % classification rate.

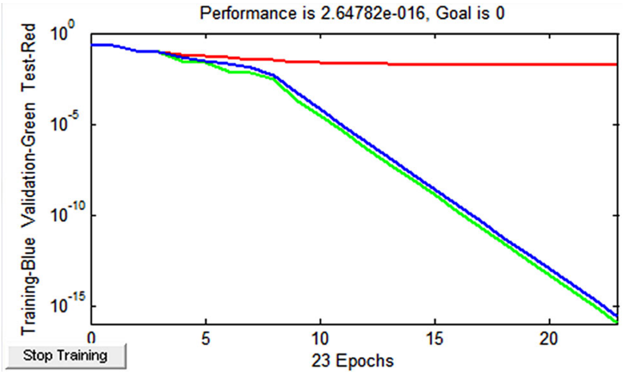

Fig. 1 Error rate curves with PLP, deltaPLP, VQLBG and FFBPNN

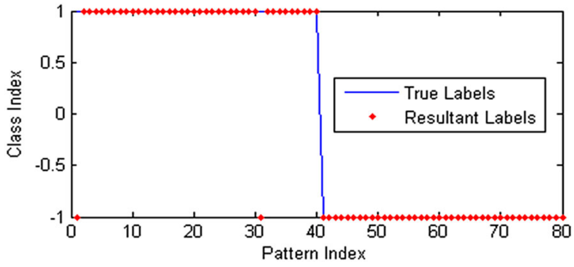

Fig. 2 Classification curves of the two dialects with PLP + ΔPLP + PCA + SVM

Table 3 Classification results of the dialects with hybrid techniques and SVM

Techniques	Classification rate (%)	Confusion matrix	Classification time (s)
PLP + ΔPLP + PCA + SVM	**97.50**	$\begin{bmatrix}0.9500 & 0.0500\\ 0 & 1.0000\end{bmatrix}$	0.073080
MFCC + ΔMFCC + PCA + SVM	90.00	$\begin{bmatrix}0.8750 & 0.1250\\ 0.0750 & 0.9250\end{bmatrix}$	0.059511
PLP + ΔPLP + VQLBG + SVM	81.25	$\begin{bmatrix}0.7500 & 0.2500\\ 0.1250 & 0.8750\end{bmatrix}$	0.059677
MFCC + ΔMFCC + VQLBG + SVM	85.00	$\begin{bmatrix}0.85000.1500\\ 0.15000.8500\end{bmatrix}$	0.067487

Bold value indicates the best performance obtained by using these techniques

Table 4 Performance evolution depending on γ and C

C	2^{1}	2^{2}	2^{3}	2^{4}	2^{5}	2^{6}	2^{7}	2^{8}	2^{9}	2^{10}	2^{11}	2^{12}
γ	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}
Perf. %	65.00	81.25	93.75	95.00	96.25	95.00	95.00	95.00	95.00	97.50	90.00	88.75

In Table 3, the classification rate, the confusion matrix and the classification time for each hybrid technique are shown.

γ and C were tested in the range indicated in the Table 4 by using hybrid techniques PLP, ΔPLP and PCA. For the other used hybrid techniques, we obtained almost the same evolution of the performances.

Here we chose $\gamma = 2^{-10} : C = 2^{10}$ because these values realized the best performance.

In Fig. 2, the result of classification after applying SVMClass is represented. The curve in blue color represents the original labels provided to SVM: labels for Tunisian dialect (the first 40 vectors) are represented by $\ll -1 \gg$ and that of Moroccan (to the second 40 vectors) by $\ll -1 \gg$. The found labels after applying SVM are represented by a red color.

The results mentioned in both previous tables prove that FFBPNN is better than SVM in term of performance with a greater computational time. The computational time was equal to 0.073080 s when using SVM and 3628.540361 s when using FFBPNN. It is worth mentioning that our system has realized satisfactory results with both classification techniques.

6 Conclusion and future work

In this paper, we proposed a Maghrebian dialect classification system based on hybrid recognition techniques for feature extraction and separately FFBPNN then SVM for classification phase. The reached classification rates were 98.3 and 97.5 % with FFBPNN and SVM respectively. FFBPNN outperformed SVM in term of performance but with a higher computational time. Moreover, we noticed that PLP followed by ΔPLP has realized the best performance with either FFBPNN or SVM. In the future, we are planning to expand our database in order to cover many ADs, extend our work for continuous AD classification and use more advanced techniques such as gammatone filter cepstral coefficients (GTCC), Gammachirp filter and Boosting Algorithm.

References

Alorifis, F. S. (2008). Automatic identification of arabic dialects using hidden markov models, Thesis, University of Pittsburgh.

Ameen, A., Uma, R., & Madhusudana, R. (2012). Speaker recognition system using combined vector quantization and discrete Hidden Markov model. *International Journal Of Computational Engineering Research, 2*(3), 2250–3005.

Amour, M., Bouhjar, A., & Boukhris, F. (2004). *Introduction to amazigh language*. Paris: IRCAM.

Antoniol, G., Rollo, V. F., & Venturi, G. (2005). *Linear Predictive Coding and Cepstrum coefficients for mining time variant information from software repositories*. St. Louis: International Workshop on Mining Software Repositories.

Barkat-Defradas, M., Hamdi, R., & Pellegrino, F. (2004). *From linguistic characterization to automatic identification of arabic dialects* (pp. 29–30). Paris: MIDL.

Biadsy, F. Hirschberg, J. & Habash, N. (2009). Spoken arabic dialect identification using phonotactic modeling. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages at the meeting of the European Association for Computational Linguistics (EACL)*, Athens.

Boril, H., Sangwan, A., Hansen, J, H.L. (2012). Arabic Dialect Identification—'Is the Secret in the Silence and Other Observations", Center for Robust Speech Systems (CRSS), Erik Jonsson School of Engineering,University of Texas at Dallas, Richardson, INTERSPEECH.

Boukous, A. (1998). The Moroccan Sociolinguistic Situation. *Plurilinguismes (Le Maroc)* (Vol. 16, pp. 5–30). Paris: Centre d'Etudes et de Recherches en Planification Linguistique.

Chang, C.-C. & Lin, C.-J. (2004). LIBSVM—a library for support vector Machines, 2004. http://www.csie.edu.tw/cjlin/libsvm/.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery, 2*, 121–167.

Cobert, D. (2003). Darija, A language of Modernity, estidios dialectlogia norteafricana y andalusi, 2003.

Dasigi, P., & Diab, M. (2011). CODACT: towards identifying orthographic variants in dialectal arabic, In *Proceedings of the 5th International Joint Conference on Natural Language Processing* (pp. 318–326), Chiang Mai, Thailand, 8–13.

Diab, M., Hacioglu, K., & Jurafsky, D. (2007). Automated methods for processing Arabic text: From Tokenization to base phrase chunking. In A. den van Bosch, A. den van Bosch, & A. Soudi (Eds.), *Arabic computational morphology: Knowledge-based and empirical methods*. New York: Springer.

Elfardy, H. & Diab, M. (2103) Sentence-level dialect identification in arabic.In *Proceedings of the 51st AnnualMeeting of the Association for Computational Linguistics*, ACL 2013, Sofia.

Gunawan, W., & Hasegawa-Johnson, M. (2001). *PLP coefficients can be quantized at 400 bps* (pp. 1–4). Salt Lake City: ICASSP.

Habash, N., Rambow, O., & Roth, R. (2009). MADA + TOKAN, A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. In K. Choukri, & B. Maegaard (Eds.) Proceedings of the second international conference on Arabic Language resources and tools. The MEDAR Consortium.

Habash, N., Roth, R., Rambow, O., Eskander, R., & Tomeh, N. (2013). Morphological Analysis and Disambiguation for Dialectal Arabic, In Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies (NAACL-HLT), Atlanta.

Hamdi, R., (2007). Rhythmic variation in arabic dialects, Thèse Université 7 novembre de Carthage Tunisie.

Hassine, M., Boussaid, L., & Messaoud, H. (2015). Hybrid techniques for Arabic Letter recognition. *International Journal of Intelligent Information Systems, 4*(1), 27–34.

Haykin, S. (2009). *Neural networks and learning machines*. New York: Prentice Hall.

Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis for speech. *The Journal of the Acoustical Society of America, 87*, 1738–1752.

Lindasalwa, M., Mumtaj, B., & Elamvazuthi, I. (2010). Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. Journal of Computing, *2*(3)

MATLAB User's Guide. Mathworks Inc., 2006.

Nour-Eddine, L., & Abdelkader, A. (2015). GMM-Based Maghreb Dialect Identification System. *Journal of Information Processing Systems, 11*(1), 22–38.

Pasha, Arfath, Al-Badrashinyy, Mohamed, Diaby, Mona, El Kholy, Ahmed, Eskander, Ramy, Habash, Nizar, et al. (2014). *MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic*. New York: Columbia University, Center for Computational Learning Systems, LREC.

Price, J. Sophomore student (2005) Design an automatic speech recognition system using maltab. In Progress report for: Chesapeake information based aeronautics consortium August 2005, University of Maryland Eastern Shore Princess Anne.

Sadat, F., Kazemi, F., & Farzindar, A. (2014). Automatic dentification of Arabic Language Varieties and Dialects in Social Media, In *Proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP)* (pp. 22–27), Dublin, August 24 2014.

Semet, G., & Treffo, G. (2002). Speech recognition based on MFCC coefficients, TIPE.

Shlens J. (2003). A tutorial on principal component analysis, Derivation, Discussion and Singular Value Decomposition.

Smola, A.J. & Scholkopf, B. (1998) A tutorial on support vector regression, Tech. rep., NeuroCOLT2 Technical Report NC2-TR-1998-030.

Srinivasan, A. (2011). Speech recognition using hidden markov model. *Applied Mathematical Sciences, 5*(79), 3943–3948.

Venkateswarlu, R.L.K., Kumari, R. V. & Vani Jayasri, G. (2011). Speech recognition using radial basis function neural network, IEEE.

Yu, H. & Kim, S. (2012). SVM tutorial: Classification, regression, and ranking. In *Handbook of Natural Computing* (pp. 479–506).