# INCORPORATION OF A LANGUAGE MODEL INTO A BRAIN COMPUTER INTERFACE BASED SPELLER THROUGH HMMs

*Çağdaş Ulaş and Müjdat Çetin*

Faculty of Engineering and Natural Sciences, Sabanci University, Orhanlı, Tuzla, 34956 Istanbul, Turkey

## ABSTRACT

Brain computer interface (BCI) research deals with the problem of establishing direct communication pathways between the brain and external devices. The primary motivation is to enable patients with limited or no muscular control to use external devices by automatically interpreting their intent based on brain electrical activity, measured by, e.g., electroencephalography (EEG). A widely studied BCI set up involves having subjects type letters based on so-called P300 signals generated by their brains in response to visual stimuli. Due to the low signal-to-noise ratio (SNR) of EEG signals, brain signals generated for a single letter often have to be recorded many times to obtain acceptable accuracy, which reduces the typing speed of the system. Conventionally the measured signals for each letter are processed and classified separately. However, in the context of typing letters within words in a particular language, neighboring letters would provide information about the current letter as well. Based on this observation, we propose an approach for incorporation of such information into a BCI-based speller through hidden Markov models (HMM) trained by a language model. We then describe filtering and smoothing algorithms for inference over such a model. Experiments on real EEG data collected in our laboratory demonstrate that incorporation of the language model in this manner results in significant improvements in classification accuracy and bit rate.

***Index Terms—*** Brian computer interface, hidden Markov model, P300 speller, language model, Viterbi algorithm

## 1. INTRODUCTION

Humans severely affected by conditions such as amyotrophic lateral sclerosis (ALS), brainstem stroke, and spinal cord injury may lose all voluntary muscle control and may be unable to communicate. The idea of brain-computer interfaces (BCIs) involves the creation of a new output channel for such individuals so the neuronal activity of the brain can be directly used to communicate with the outside world. Studies over the last two decades have shown that non-invasively obtained electrical signals through the scalp-recorded electroencephalogram (EEG) can be used as the basis for BCIs. In an EEG-based BCI system, incoming signals from an EEG amplifier are processed and classified to decode the user's intent [1]. Current studies allow the users to perform several actions: controlling robot arms [2, 3], selecting and typing letters on a screen [4, 5] or moving a cursor [6].

The P300 speller is one of the most common types of BCI that enables a subject to write text on a computer screen. The P300 speller paradigm was introduced by Farwell and Donchin in [4]. P300 is an event related potential (ERP) that occurs in the brain as a response to a visual or auditory stimulus. Columns and rows of a matrix of characters (See Fig. 1) flash randomly as the subject attends to one character. The brain is expected to generate a P300 response for the flashes containing the attended character. Due to the low SNR and variability of EEG signals, P300-based BCI typing systems need several number of stimulus repetitions to increase classification accuracy, which causes low symbol rates [7, 8]. Various aspects of the P300 speller were examined to improve the performance such as electrode selection, stimulus shape and dimension, different flashing paradigms [9] and several signal processing and classification methods [10, 11]. However, the idea of integration of a language model into the decision making algorithm to predict the current letter using the previous letters is not common. Speier et al. [12] proposed a natural language processing (NLP) approach which exploits the classification results on the previous letters to predict the current letter based on learned conditional probabilities. Orhan et al. [7] created a system using a non-conventional flashing paradigm, the RSVP keyboard, and merged the context-based letter probabilities and EEG classification scores by using a recursive Bayesian approach. Both of these ideas showed that integrating information about the linguistic domain can improve the speed and accuracy of a BCI communication system.

In this paper, we propose a new approach for the integration of a language model and the EEG scores based on a second order Hidden Markov Model (HMM). We use Forward-Backward and Viterbi algorithms to make decisions on the letters typed by the subjects. There have been several works on HMM applications of BCI-based contexts, such as in [11]. However, using HMM based on a language model as we propose has not been practised before. Our work is significantly different from previous work in [7, 12]. The approach in [12] is greedy in the sense that the prediction for the current letter is performed conditioned only on the letters declared by the system for the previous time instants. On the other hand, our approach is fully probabilistic. It acknowledges that previous decisions contain uncertainties as well, and performs prediction by considering the computed probabilities of all letters in the previous instant(s), rather than just the declared ones. Both [7] and [12] exploit information in the previous letters for the current letter. In contrast, our approach takes advantage of both the past and the future. In this way, previously declared letters can be updated as new information arrives. We present experimental results based on EEG data collected in our laboratory through P300-based spelling sessions. We consider both the original

This work was partially supported by the Scientific and Technological Research Council of Turkey under Grant 11E05 and through a graduate fellowship, and by Sabanci University under Grant IACF-11-00889. We would like to thank Armağan Amcalar for sharing his code for pre-processing and the BLDA classifier as well as the stimulus software.

measurements, as well as their noisy versions to test the robustness of the approach to reductions in SNR. Our results show that the speed and the classification accuracy of the BCI system can be improved by using the proposed approach in both noiseless and the noisy case.

## 2. METHODS

The stimulus software used during EEG data acquisition and data pre-processing methods used in this study were described in detail in [13, 14]. The classification algorithm is composed of two steps: 1. the Bayesian Linear Discriminant Analysis (BLDA) classifier is used to calculate classification scores for each letter in the sequence independently, 2. These scores are integrated into a HMM and with the help of a trigram language model, the classifier decides on each letter in the sequence by using Forward-Backward and Viterbi algorithms. Following sections will provide the details of this approach.

### 2.1. Bayesian Linear Discriminant Analysis (BLDA)

The first step of our classification approach involves applying BLDA on the EEG data. The detailed explanation of BLDA can be found in [15]. The classification problem here involves two classes: whether an epoch (EEG data corresponding to a single flash) in the test data contains the attended character or a non-attended character. In order to investigate this, the epochs in the training data are assigned labels based on these two classes. Then, BLDA calculates a score for each epoch of test data, reflecting its similarity to the attended class.

The score for each character can be found by summing the individual scores for two flashes that contain the corresponding character. Scores are added up in consecutive repetitions of stimuli (called trial groups) for typing a particular character. The classifier chooses the character with the maximum score. In our work, we use the scores, rather than the classification decisions of BLDA.

### 2.2. Language Model-based BCI

We believe that combining the BLDA scores with conditional probabilities for characters based on a language model can lead to performance improvements in BCI-based spelling. Therefore, we propose to construct an HMM where each symbol in the speller matrix forms the latent variable and BLDA scores of all symbols corresponding to a run (all trial groups for typing a character) form the observed variable.

#### *2.2.1. Forward-Backward Algorithm*

Let $S_t$ denote the state at time *t* where $t \in \{1,2,....,T\}$ and let us define an observation sequence, $O = O_1 O_2 .....O_T$ , where each $O_k$ represents the BLDA scores of all possible symbols for *k*th letter (run) of the target word. The forward-backward algorithm first computes a set of forward probabilities for all $t \in \{1,2,....,T\}$, which define the probability of partial observation sequence until time *t* and being in state *i* at time *t*, where each *i* is an element of speller matrix. At the second step, the algorithm computes backward probabilities providing the probability of the partial observation sequence from to *t+1* to *T*, given the state *i* at time *t*. Then, we can combine these two sets of probabilities to calculate the probability distribution over states at any particular time ,

$$P(S_t = i \mid O_{1:T}) \propto P(O_{1:t}, S_t = i) P(O_{t+1:T} \mid S_t = i) \qquad (1)$$

where the first term on the right-hand side stands for forward probability at time $t$ and second term stands for backward probability at time $t$ denoted as respectively, $\alpha_t(i)$ and $\beta_t(i)$ .

Let us consider a second order HMM. In this case $\alpha_t(i)$ and $\beta_t(i)$ can be recursively computed as follows [16]:

$$\alpha_1(i) = P(S_1 = i) P(O_1 \mid S_1 = i) \qquad (2)$$

$$\alpha_2(j) = \sum_i \alpha_1(i) P(S_2 = j \mid S_1 = i) P(O_2 \mid S_2 = j) \qquad (3)$$

$$\alpha_t(k) = \sum_j \sum_i \alpha_{t-1}(i,j) a_{ijk} P(O_t \mid S_t = k) \qquad (4)$$

where $a_{ijk} = P\left(S_t = k \mid S_{t-1} = j, S_{t-2} = i\right)$, $3 \le t \le T$ , and each $i,j,k$ is an element of the matrix. In a similar manner,

$$\beta_T(i) = 1 \qquad (5)$$

$$\beta_t\left(i\right) = \sum_j \sum_k \beta_{t+1}\left(j,k\right) a_{ijk} P(O_{t+1} | S_t = k) \qquad (6)$$

for $T-1 \ge t \ge 1$ .

Assuming all BLDA epoch scores of a run are conditionally independent given the class labels, we can compute $P(O_t \mid S_t = i)$ for each $t \in \{1,2,....,T\}$ and for any number of available trial groups $N$ as follows:

$$P(O_t \mid S_t = k) = \prod_{n=1}^{N} p(O_t(x_k, n)) (\prod_{n=1}^{N} \prod_{k'} p(O_t(x_{k'}, n))) \qquad (7)$$

where $O_t\left(x_k, n\right)$ represents the epoch scores containing the character $k$ and $O_t\left(x_{k'}, n\right)$ represents those without character $k$ . Given the class label $l_k$ , we have observed that $p(O_t(x_k, n) \mid l_k)$ and $p(O_t(x_{k'}, n) \mid l_{k'})$ are normally distributed by analyzing the distribution of the training data scores. We have estimated the parameters of Gaussian densities to the test epoch scores for both attended and not-attended classes from training data scores, where $l_k = 1$ for attended epochs and $l_{k'} = -1$ for non-attended epochs in the training set.

The initial probability, $\pi_i = P\left(S_1 = i\right)$ , and the transition probabilities, $a_{ij} = P\left(S_2 = j \mid S_1 = i\right)$ and $a_{ijk}$ , are estimated using a trigram language model [17], which corresponds to a second-order HMM. Trigrams for the Turkish language were obtained from a translation of a book which contains more than 300,000 words including various types of lexicon. Laplace smoothing technique was used for assigning non-zero probabilities for unseen n-grams [18].

Having calculated the forward and backward probabilities based on (4) and (6), the probability of being in state *i* at time *t* given the observation sequence *O* can be expressed as follows [19]:

$$P(S_t = i \mid O_{1:T}) = \gamma_t\left(i\right) = \frac{\alpha_t(i)\beta_t(i)}{\sum_i \alpha_t(i)\beta_t(i)} \qquad (8)$$

**Table 1.** Average performance values for different methods averaged over experiments considering {1,2,3,4,5} trial groups. The forward, forward-backward and Viterbi methods are based on the proposed model.

| Method | Average Values Without Noise | | Average Values With Noise $\sigma = [0, 1]$ | |
|---|---|---|---|---|
| | Classification Accuracy (%) | Bit rate (bits/min) | Classification Accuracy (%) | Bit rate (bits/min) |
| **BLDA** | 56.66 | 15.38 | 50.93 | 12.71 |
| **NLP** | 60.90 | 17.15 | 55.12 | 14.43 |
| **Forward** | 72.30 | 22.90 | 66.91 | 19.78 |
| **Forward- Backward** | 74.10 | 22.78 | 68.73 | 20.57 |
| **Viterbi** | 76.15 | 23.88 | 70.15 | 20.83 |

By using (8), we can estimate the individually most likely state or character at any time *t*,

$$\tilde{S}_t = \arg\max_i[\gamma_t(i)] \quad \text{for } 1 \le t \le T.$$

*2.2.2. Viterbi Algorithm*

The forward-backward algorithm can be used to determine the most likely character for any *k*th letter of a target word. However, it cannot be helpful to find the most likely letter sequence for a given model. In order to find the single best letter sequence, we use the Viterbi algorithm [20] on our proposed HMM. The required state transition probabilities and observation symbol probabilities for this algorithm were already provided in Section 2.2.1. Given multiple-trial EEG data for each letter in the sequence, the Viterbi algorithm produces the most probable letter sequence of a corresponding target word.

## 3. EXPERIMENTAL RESULTS

In this study, six healthy subjects, whose ages varied between 18 and 30, took part in an offline spelling. Only two of the subjects had previous BCI experience. The system used the most popular stimulus type, a $6 \times 6$ matrix of characters. The rows and columns of the matrix are highlighted in a block-randomized fashion; i.e., in 12 flashes, each row and column is flashed exactly once with an ISI of 125 ms and a flash duration of 50 ms. Each subject underwent two sessions: one being the training session and the other, the test session. The training session of each subject featured 14 runs (characters) with 2 Turkish words. The test sessions featured 26 runs with 4 Turkish words. All six words chosen for typing in training and test sessions are different from each other. The classifier was trained on the first session and tested on the second. A pre-determined number of trial groups make up a run. In this study, the maximum number of the trial groups was set to 15. The data were recorded in BioSemi ActiView software and MATLAB was used for offline data analysis and the classification process.

The performance evaluation of our P300 based BCI system depends on two important criteria: accuracy and bit rate. Accuracy is calculated by dividing the total number of correct classifications for each character in a session by the total number of classifications. The bit per symbol, $B$, is also computed as in [1]:

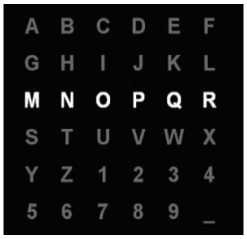
$$B = \log_2 N + P\log_2 P + (1-P)\log_2 \frac{1-P}{1-N}$$

where $P$ is the accuracy of the classification and $N$ is the number of characters in the speller matrix given in Figure 1. The multiplication of $B$ by the number of symbol selections per minute gives the bit rate, in bits per minute. Since one trial group takes 1.5 s and 3.5 s is needed to display the target letter to the subject, there can be 12 characters at maximum that a subject can manage to type in a minute. Hence, the maximum bit rate of our system using a perfect classifier for offline classification is 62.04 bits/min.

Five different methods for classification analysis are compared in this study: a general BLDA method that does not use any type of language modeling for letter prediction, the "NLP" method that develops a language model which only depends on the integration of the EEG scores at the current time with character trigram probabilities based on decisions of the previous time and forward, forward-backward and Viterbi methods that use a language model by constructing proposed HMM described in Section 2.2.

All methods are implemented for two cases: first one is using the original data obtained from the offline experiment and second one is using the new scores obtained by adding Gaussian noise with various standard deviations to the BLDA scores of the original data. The underlying reason for performing the second case is to test the success and effectiveness of language modeling in adverse conditions such as presence of noise.

Since speed is very important for the effectiveness of real-time BCI communication, we first present performance values averaged over experiments using {1,2,3,4,5} trial groups (repetitions) only, rather than considering up to 15 trial groups. This provides a comparison of methods in the high-speed regime. Our later results will involve more trial groups. The average classification accuracy and bit-rate values are shown in Table 1. The results are quite promising. All three methods based on our model significantly



| | | | | | |
|---|---|---|---|---|---|
| A | B | C | D | E | F |
| G | H | I | J | K | L |
| M | N | O | P | Q | R |
| S | T | U | V | W | X |
| Y | Z | 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 | 9 | _ |

**Fig. 1.** The speller matrix used in this study. '_' denotes space.

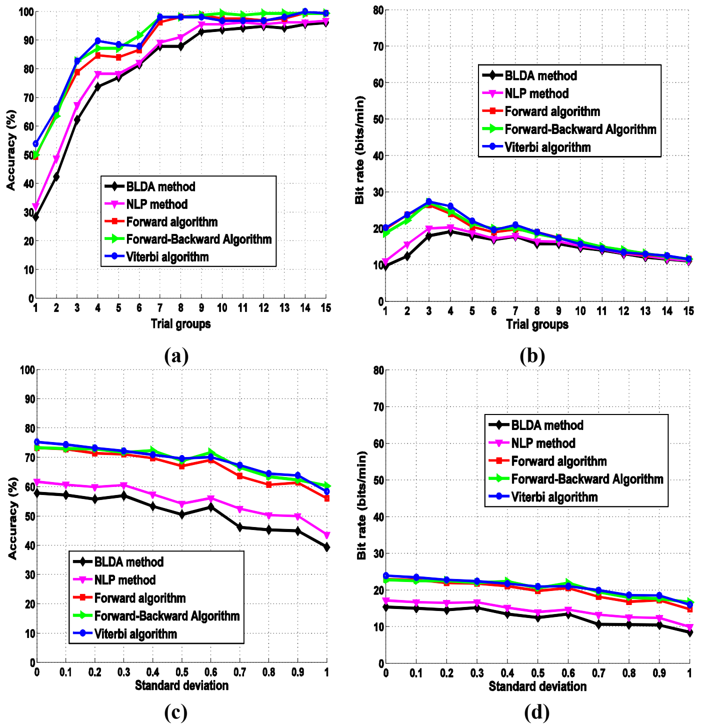


**Fig. 2.** Average performance in the noiseless case (top) and the noisy case (bottom). (a),(b) Accuracy and bitrate versus the number of trial groups. (c),(d) Accuracy and bitrate versus noise standard deviation averaged over using {1,2,3,4,5} trial groups.

improve the speed and accuracy of the BCI system compared to the BLDA method and the NLP method of [12] both in the presence of noise and in the non-noisy case.

Results in Table 1 also suggest that the impact of the proposed methods on both accuracy and bit-rate is increased in the presence of noise compared to non-noisy case. To be more precise, while in the non-noisy case, the overall improvement from BLDA method to the Viterbi method is 34.4% and 55.2% for accuracy and bit rate, respectively, the improvement is increased to 37.8% and 63.9% in the presence of noise. For the forward-backward method, these values increased from 30.7% and 48.1% to 34.9% and 61.8%. Figure 2 (a) and (b) show the average accuracy and bit rate values of all 6 subjects for each number of trial groups considered (max=15). In the noiseless case, the BLDA method on average needs at least 7 trial groups to reach 90% accuracy while the Viterbi method can achieve this after 4 trial groups. Since one trial group lasts 1.5 s, our approach will predict the target letter with 90% accuracy 4.5 s earlier than BLDA does. Figure 2(b) illustrates the remarkable effect of our model on the speed of the BCI system particularly in the first three trial groups.

Compared with the NLP method proposed in previous work [12], our system can achieve both higher accuracy and speed. Table 1 asserts that in the non-noisy case, the accuracy and bit rate are improved 25% and 39.2%, respectively, by the Viterbi method of the proposed HMM. This significant difference arises from the fact that if an error is made in the selection of previous letters, then the classifier will decide on the current letter just based on this wrong letter. However, our model keeps the all possible symbol probabilities of the previous time and these will be taken into account when estimating the current letter.

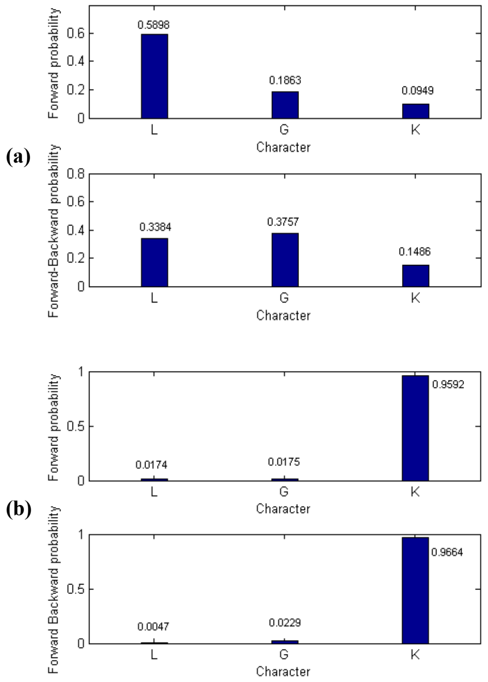


**Fig. 3.** (a) Forward and Forward-Backward (F-B) probabilities for the most probable three letters in the first trial group (b) Forward and F-B probabilities for the most probable three letters in the second trial group. The actual target letter of the run is 'K'.

Based on our results, the performances of the forward algorithm and forward-backward algorithm are close to each other. The reason behind this can be explained by the important role of BLDA scores on decision making. If a relatively low probability score is assigned to the target (correct) letter by BLDA, then the trigram probabilities obtained from the corpus may not manage to maximize this low probability among all the other symbol probabilities. Even if the forward-backward algorithm (smoothing) gives better probability scores than the forward algorithm (filtering) does, this improvement may not be reflected in classification accuracy. An illustration of this issue is given in Figure 3 for a data sample. Although the target letter's probability is increased in the first trial group by smoothing, the increment is not enough to maximize this probability. Therefore, the classifier will choose the wrong letter. This brief example aims to show how information from the past and the future letters are reasonably incorporated in our approach, although some of this effect may be invisible in gross classification results.

## 4. CONCLUSION

In this paper, we have developed a new P300 based BCI system incorporating a language model constructed by an HMM and demonstrated the performance of our proposed model. We have performed offline experiments with 6 healthy subjects. Our results show that the proposed model can achieve higher speed and accuracy compared to relevant recent work. We have also shown that the impact of our language model is even greater in the presence of noise. This demonstrates that the proposed model will exhibit robustness to the potentially poor conditions of a data collection procedure.

## REFERENCES

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain-computer interfaces for communication and control," *Clin Neurophysiol*, vol. 113, pp. 767-91, Jun. 2002.

[2] G. R. Muller-Putz, R. Scherer, G. Pfurtscheller, and R. Rupp, "EEG-based neuroprosthesis control: a step towards clinical practice," *Neurosci Lett,* vol. 382, pp. 169-74, Jul. 2005.

[3] R. Lauer, P. Peckham, K. Kilgor, and W. Heetderks, "Applications of cortical signals to neuroprosthetic control: a critical review," *IEEE Trans. Rehabil. Eng,* vol. 8, no. 2, pp. 205-8.

[4] L. A. Farwell and E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalogr Clin Neurophysiol,* vol. 70, pp. 510-23, Dec. 1988.

[5] B. Obermaier, G. Müller, and G. Pfurtscheller, " "Virtual keyboard" controlled by spontaneous EEG activity," *IEEE Trans Neural Syst Rehabil Eng,* vol. 11, pp. 422-6, Dec. 2003.

[6] J. R. Wolpaw and D. J. McFarland, "Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans," *Proc Natl Acad Sci U S A,* vol. 101, pp. 17849-54, Dec. 2004.

[7] U. Orhan, D. Erdogmus, B. Roark, B. Oken, S. Purwar, K. E. Hild, A. Fowler, and M. Fried-Oken, "Improved accuracy using recursive Bayesian estimation based language model fusion in ERP-based BCI typing systems," *34th Annual International IEEE EMBS Conference of the Engineering in Medicine and Biology Society*, San Diego, California, 2012.

[8] U. Orhan, K.E. Hild, D. Erdogmus, B. Roark, B. Oken, and M. Fried-Oken, " RSVP keyboard: An EEG based typing interface," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).* 2012, pp. 645-648.

[9] G. Townsend, B. LaPallo, C. Boulay, D. Krusienski, G. Frye, C. Hauser, N. Schwartz, T. Vaughan, J. Wolpaw, and E. Sellers, "A novel P300-based brain–computer interface stimulus presentation paradigm: moving beyond rows and columns," *Clin. Neurophysiol.* 121 1109–20 , 2010.

[10] H. Serby, E. Yom-Tov, and G. F. Inbar, "An improved P300-based brain-computer interface," *IEEE Trans Neural Syst Rehabil Eng,* vol. 13, pp. 89-98, Mar. 2005.

[11] B. Obermaier, C. Guger, C. Neuper, and G. Pfurtscheller, " Hidden Markov models for online classification of single trial EEG data*," Pattern Recognit. Lett,* 22 , 1299–309.

[12] W. Speier, C. Arnold, J. Lu, R.K. Taira, and N. Pouratian, "Natural language processing with dynamic classification improves P300 speller accuracy and bit rate," *J. Neural Eng.*, 9(1):016004 , Feb. 2012.

[13] A. Amcalar and M. Çetin, "Design, implementation and evaluation of a real-time P300-based brain-computer interface system," *International Conference on Pattern Recognition (ICPR)*, Istanbul, Turkey, August. 2010.

[14] A. Amcalar, "Design, implementation and evaluation of a real-time P300-based brain-computer interface system," Master thesis, Sabanci University, Feb. 2010.

[15] U. Hoffmann, G. Garcia, J. Vesin, and T. Ebrahimi, "Application of the evidence framework to brain-computer interfaces," *IEEE Eng Med Biol Soc (EMBC),* vol. 1, pp. 446-9, 2004.

[16] A. Kriouile, J. F. Mari, and J. P. Haton,"Some Improvements in Speech Recognition Algorithms based on HMM," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Albuquerque, pp. 545-548, 1990.

[17] F. J. Damerau, *Markov Models and Linguistic Theory*. Mouton. The Hague, 1971.

[18] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, 2009.

[19] L. R. Rabiner, "A Tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE***,** vol. 77**,** no. 2, pp. 257-286**,** Feb. 1989.

[20] Y. He, "Extended Viterbi algorithm for second-order hidden Markov process," *Proceedings of the IEEE 9th International Conference on Pattern Recognition (ICPR)*, pp. 718- 720, 1988.