INCORPORATION OF A LANGUAGE MODEL INTO A BRAIN COMPUTER INTERFACE BASED SPELLER

by

Çağdaş Ulaş

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University
Spring 2012-2013

INCORPORATION OF A LANGUAGE MODEL INTO A BRAIN COMPUTER INTERFACE BASED SPELLER

APPROVED BY

Assoc. Prof. Dr. Müjdat ÇETİN .....
(Thesis Supervisor)

Assoc. Prof. Dr. Berrin YANIKOĞLU .....

Assoc. Prof. Dr. Volkan PATOĞLU .....

DATE OF APPROVAL: 13.09.2013

to my family and my love Beşiktaş

"You will never be happy if you continue to search for what happiness consists of. You will never live if you are looking for the meaning of life."
—*Albert Camus*

Acknowledgments

I wish to express my gratitude to my supervisor, Müjdat Çetin , whose expertise, understanding, and patience, added considerably to my graduate experience. I am grateful to him not only for the completion of this thesis, but also for her unconditional support during my Master education.

I wish to express my appreciation to Armağan Amcalar for his guidance in my senior project and his unconditional collaboration and support for my Master thesis.

I would like to give my special thanks to Hakan Erdoğan for his valuable guidance and support both on my Master courses and thesis study.

I also would like to thank TÜBİTAK for providing the necessary financial support for my graduate education[1].

I am grateful to my committee members Berrin Yanıkoğlu and Volkan Patoğlu for taking the time to read and comment on my thesis.

I owe special thanks to all my friends and colleagues in the lab for their friendship and assistance during my Master study. I also would like to thank all the subjects for their pleasure to participate experiments by sacrificing their spare time.

[1]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

Finally, I would like to thank my family for their valuable supports, love and belief in me.

INCORPORATION OF A LANGUAGE MODEL INTO A BRAIN COMPUTER INTERFACE BASED SPELLER

Çağdaş Ulaş

EE, M.Sc. Thesis, 2013

Thesis Supervisor: Müjdat Çetin

Keywords: Brain-computer interface, electroencephalography, Hidden Markov Model, P300 speller, n-gram language modelling

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). The P300 speller is a widely practised BCI set up that involves having subjects type letters based on P300 signals generated by their brains in response to visual stimuli. Because of the low signal-to-noise ratio (SNR) and variability of EEG signals, existing typing systems use many repetitions of the visual stimuli in order to increase accuracy at the cost of speed. The main motivation for the work in this thesis comes from the observation that the prior information provided by both neighbouring and current letters within words in a particular language can assist letter estimation with the aim of developing a system that achieves higher accuracy and speed simultaneously. Based on this observation, in this thesis, we present 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 in conjunction with n-gram language models for inference over such a model. We have designed data

collection experiments for offline and online decision-making which demonstrate that incorporation of the language model in this manner results in significant improvements in letter estimation and typing speed.

DİL MODELİ DESTEKLİ BİR BEYİN-BİLGİSAYAR ARAYÜZÜ TABANLI HECELETİCİ

Çağdaş Ulaş

EE, Yüksek Lisans Tezi, 2013

Tez Danışmanı: Müjdat Çetin

Anahtar Kelimeler: Beyin-Bilgisayar Arayüzü, elektroensefalografi, Saklı Markov Modeli, P300 heceleticisi, n-gram dil modeli

Özet

Beyin-Bilgisayar Arayüzü (BBA) araştırmaları, beyin ve dış aygıtlar arasında doğrudan iletişim kanalı kurma sorunu ile ilgilenmektedir. Buradaki birincil motivasyon, sınırlı derece kas kontrolüne sahip olan veya hiç sahip olmayan hastaların elektroensefalografi (EEG) gibi yöntemlerle beyin elektriksel aktivitelerini ölçerek otomatik olarak niyetlerini yorumlayıp onların dış aygıtlar kullanmasına olanak sağlamaktır. Yaygın olarak üzerinde uygulamaların gerçekleştirildiği BBA düzeneklerinden birisi olan P300 heceleticisi, kullanıcıların öngörülemeyen uyaranlara karşı beyinlerinde cevap olarak oluşan ve P300 diye bilinen sinyallere dayalı bir şekilde harf yazmalarını içerir. EEG sinyallerinin düşük sinyal-gürültü oranı ve çeşitliliği nedeniyle, mevcut heceleme sistemleri, hız pahasına başarım değerini arttırmak için fazla sayıda uyaran tekrarlamasını kullanmaktadır. Bu tezdeki çalışmaya motivasyon sağlayan temel gözlem, belirli bir dildeki kelimeler içinde yer alan komşu ve mevcut harfler tarafından sağlanan önsel bilginin, aynı anda daha yüksek başarım ve hız değerlerinin sağlandığı bir sistemin geliştirilmesinde yardımcı olabileceğidir. Bu gözleme dayanarak, mevcut tez çalışmasında, bir dil modeli tarafından eğitilmiş Saklı Markov Modeli (SMM) yapısı aracılığıyla BBA tabanlı heceleticinin içine bu önsel bilgilerin dahil edildiği bir yaklaşım sunuyoruz. Böyle bir model üzerinde çıkarsama yapmak

için n-gram dil modeliyle bağlantılı olarak kullandığımız filtreleme ve yumuşatma algoritmalarını tanımlıyoruz. Çevrimdışı ve çevrimiçi karar verme üzerine tasarladığımız veri toplama deneyleri, dil modelinin bu şekilde karar sürecine dahil edilmesinin harf tahmini doğruluğunda ve heceleme hızında önemli iyileştirmelere yol açtığını gösteriyor.

Table of Contents

Acknowledgments **v**

Abstract **vii**

Özet **ix**

1 Introduction **1**
1.1 Scope and Motivation . . . 2
1.2 Contributions . . . 4
1.3 Thesis Outline . . . 5

2 Background on BCI and P300 Spellers **7**
2.1 Introduction . . . 7
2.2 Electroencephalography (EEG) . . . 8
2.2.1 Electrodes . . . 9
2.3 A General BCI System . . . 10
2.3.1 Motor Imagery . . . 12
2.3.2 Event Related Potentials . . . 13
2.4 P300 based BCI systems . . . 15
2.4.1 Data processing procedure for a P300-based BCI system . . . 18
2.5 P300 Speller classification techniques . . . 19
2.6 Language Model . . . 22
2.7 Summary . . . 22

3 Language Model-based P300 Speller **24**
3.1 Bayesian Linear Discriminant Analysis (BLDA) . . . 24
3.2 Logistic Regression . . . 28
3.2.1 Estimating Logistic Regression Parameters . . . 29
3.2.2 Regularization in Logistic Regression . . . 31
3.3 Language Model-based BCI . . . 32
3.3.1 Forward-Backward Algorithm . . . 33
3.3.2 Viterbi Algorithm . . . 36

3.3.3 N-gram Probabilities and Katz Back-off Smoothing 36
3.4 Channel Selection 38
3.5 Comparison with Relevant Works 39

4 Offline Analysis 41
4.1 Background 41
4.2 Terminology 42
4.3 P300 Classification Problem 43
4.4 Methods 44
4.4.1 Data pre-processing 44
4.4.2 Classification 46
4.5 Experiments 47
4.5.1 Datasets 47
4.5.2 Experimental Results 48

5 Online Spelling Experiments 67
5.1 Background 67
5.2 Method 68
5.2.1 Data pre-processing 68
5.2.2 Classification 69
5.3 Results 70

6 Conclusions and Future Work 73

Bibliography 77

List of Figures

1.1 The speller matrix used in this study. "_ " denotes space. 3

2.1 64-channel electrode cap using international 10-20 system for electrode distribution. Taken from [29]. 9

2.2 Active electrode sets used in this study. Taken from [32]. 10

2.3 Electrode placement layout according to 10-20 electrode system. Taken from [29]. 11

2.4 Illustration of a typical BCI system. Taken from [34]. 12

2.5 Average of brain signals over trials following a visual stimuli obtained from the central zero (Cz) electrode. The blue dashed line is the average response of trials where a P300 wave is visible, the solid red line shows the average response of trials where no P300 wave is elicited 14

2.6 First P300 speller paradigm used by Donchin 16

2.7 Two different flashing paradigms (a) Hex-o-Spell interface, (b) RSVP interface 16

2.8 A screenshot of the BCI 2000 P300 speller application. *Text To Spell* indicates the pre-defined target letters. The speller will analyze evoked responses and will append the selected letter to *Text Result*. Taken from [46]. 17

2.9 A screenshot of the SU-BCI P300 Speller before the beginning of the session. 17

2.10 Optimal hyperplane for support vector machine with two classes. Taken from [53]. 21

3.1 The system diagram of the proposed P300 recognition system in this study. 25

3.2 A sequential HMM 32

3.3 BLDA score distributions: histograms of the attended (solid curve) and non-attended (broken curve) scores from BLDA. 35

4.1 Offline analysis results for subject 1. (a) Accuracy and (b) Bit-rate versus the number of trial groups. 53
4.2 Offline analysis results for subject 2. (a) Accuracy and (b) Bit-rate versus the number of trial groups. 54
4.3 Offline analysis results for subject 3. (a) Accuracy and (b) Bit-rate versus the number of trial groups. 54
4.4 Offline analysis results for subject 4. (a) Accuracy and (b) Bit-rate versus the number of trial groups. 55
4.5 Offline analysis results for subject 5. (a) Accuracy and (b) Bit-rate versus the number of trial groups. 55
4.6 Offline analysis results for subject 6. (a) Accuracy and (b) Bit-rate versus the number of trial groups. 56
4.7 Offline analysis results for subject 7. (a) Accuracy and (b) Bit-rate versus the number of trial groups. 56
4.8 Average classification performance over 7 subjects using fourgram language model with the BLDA classifier. (a) Accuracy and (b) Bit-rate versus the number of trial groups. 57
4.9 Average classification performance over 7 subjects using fourgram language model with the LR classifier. (a) Accuracy and (b) Bit-rate versus the number of trial groups. 57
4.10 Average classification performance over 7 subjects using trigram language model with the BLDA classifier. (a) Accuracy and (b) Bit-rate versus the number of trial groups. 58
4.11 Average classification performance over 7 subjects using trigram language model with the LR classifier. (a) Accuracy and (b) Bit-rate versus the number of trial groups. 58

4.12 (a) Average accuracy and (b) Average bitrate versus the number of trial groups for different n-grams using the Forward algorithm with the BLDA classifier. 60
4.13 (a) Average accuracy and (b) Average bitrate versus the number of trial groups for different n-grams using the F-B algorithm with the LR classifier. 61
4.14 (a) Average accuracy and (b) Average bitrate versus the number of trial groups for different n-grams using the Viterbi algorithm with the BLDA classifier. 61
4.15 (a)-(b) Average accuracy and bit-rate versus the number of electrodes obtained in the first 3 trial groups,(c)-(d) Average accuracy and bit-rate versus the number of electrodes obtained in the first 5 trial groups. 63
4.16 BCI Competition Dataset II offline analysis results. (a) Accuracy and (b) Bit-rate versus the number of stimulus repetitions. 65

List of Tables

4.1 Target words of our own Dataset. 47

4.2 Target words of BCI Competition II Dataset IIb. 48

4.3 Performance values for each subject obtained in the first 3 trial groups using BLDA in all approaches. 50

4.4 The resulted p-values when using BLDA classifier in all approaches. 51

4.5 Performance values for each subject obtained in the first 3 trial groups using LR in all approaches. 52

4.6 The resulted p-values when using LR classifier in all approaches. 53

4.7 Average performance values for different n-grams obtained in the first 5 trial groups using the BLDA classifier 59

4.8 Average performance values for different n-grams obtained in the first 5 trial groups using the LR classifier 60

4.9 Performance values in literature for BCI Competition Dataset II 66

5.1 Online Performances for each subject 72

Chapter 1

Introduction

People devastated by severe neuromuscular diseases, such as Amyotrophic Lateral Sclerosis (ALS), high spinal cord injuries, or brainstem strokes, share the possible ultimate fate of the "locked-in" syndrome, in which cognitive function is maintained, but voluntary movement and communication abilities are impaired [1]. Brain-computer interfaces (BCIs) is one of the most promising technologies that 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.

Currently, there are several technologies to acquire the brain signals either invasively or non-invasively. This includes techniques such as Electroencephalography (EEG) [2], Magnetoencephalography (MEG) [3], Functional Magnetic Resonance Imaging (fMRI) [4], positron emission topography (PET) [5], functional near infrared spectroscopy (fNIRS)[6], and so on.

Among them, the most widely used technique in BCI settings is EEG. EEG is a noninvasive technique that records electrical brain activity via electrodes attached to the scalp of a subject. 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 [7]. Current studies allow the users to perform several actions: controlling robot arms [8, 9], selecting and typing letters on a screen [10, 11] or moving a cursor [12].

1.1 Scope and Motivation

This thesis focuses on one of the widely studied BCI applications that enables the subjects to select characters from a matrix presented on a computer screen by analyzing and classifying EEG signals. This application is known as the P300 Speller and was first introduced by Farwell and Donchin in 1988 [13]. P300 is an event-related potential. Event-related potentials (ERPs) are involuntary stereotyped electrophysiological responses to sensory stimuli such as sound, light, electrical stimulation of the skin. The ERPs are characterized by the time after the stimuli and a positive or negative deflection of the signal. P300 is an event related potential that occurs as a response in the presence of rare external stimuli [14]. Groups of characters in a matrix grid (Figure 1.1) are flashed randomly as the subject attends one character and the flashes containing the attended character will elicit an evoked response called P300. A pattern recognition algorithm then classifies EEG responses based on features differentiating attended and non-attended flashes among the rows and among the columns and selects the character that falls in the intersection of the groups with a positive response [15]. However, the use of non-invasive BCI techniques on letter-by-letter typing systems suffers from low information transfer rate because of the necessity of repeating the same stimulus several times in order to achieve satisfactory classification accuracy, which is mainly caused by the low SNR of EEG signals and the variability of background brain activity [16,17]. Several aspects of the P300 speller have been studied for improving the information transfer rate, including various signal classification methods such as support vector machines (SVMs) [18], stepwise linear discriminant analysis (SWLDA) [19], and independent component analysis (ICA) [20]; different speller matrix sizes [21], flashing patterns [22], and inter-stimulus intervals [23].

Along with all the techniques implemented to tackle the low information rate problem of BCI communication systems, we hypothesize that language specific prior information directly integrated into the decision making algorithm can increase the speed and accuracy of the system. Although this idea has not been very common in the BCI community and most of the existing analyses have treated character selections as independent ele-

ments chosen from the speller matrix with no prior information, recently several studies that use prior knowledge coming from a particular language domain directly integrated into the letter prediction algorithm have emerged. Speier et al. [15] 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. [16] 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. Martens et al. [24] performed discriminative training on real speller data to show how decoding performance improves in conjunction with unigram letter frequency information and using a more realistic graphical model for dependencies between the brain signals and the stimulus events. Kindermans et al. [25] proposed a set of unsupervised hierarchical probabilistic models that tackle the warm-up period and stimulus repetitions problems simultaneously by incorporating prior knowledge from two sources: information from other training subjects through transfer learning and information about the words being spelled through language models. All of these ideas showed that integrating information about the linguistic domain can improve the speed and accuracy of a BCI communication system.

In this thesis, we present a new approach for the integration of a language model and the EEG scores based on a Hidden Markov Model (HMM). We use Forward-Backward and Viterbi algorithms applied on two different classification methods to make decisions on the letters typed by the subjects. We present experimental results based on EEG

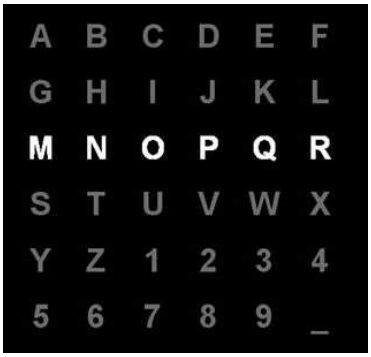


Figure 1.1: The speller matrix used in this study. "_ " denotes space.

data collected in our laboratory through P300-based offline and online spelling sessions. This study considers HMMs based on n-gram language modelling for different values of n and compares the resulting performance. The robustness of the proposed method is also tested when only data obtained by a limited number of channels is available. The results demonstrate that the speed and the classification accuracy of the BCI system can be improved by using the proposed approach in all of these cases.

1.2 Contributions

As it was mentioned before, the use of noninvasive BCI techniques on letter-by-letter spelling systems suffers from low accuracies for symbol selection due to low signal to noise ratio and variability of background brain activity. Hence, several stimulus repetitions (several trials) are required to obtain an acceptable accuracy in P300 signal classification. Additionally, it is difficult to design a perfect classifier for all subjects because of the subject variability problem. In other words, the performance of the designed system is highly affected by the physical and mental condition of a subject which leads to subject-specific problems in BCI. In this thesis, we aim to utilize the natural language information as a prior in our decision-making algorithm to improve the speed of the BCI system as well as the accuracy since this increases the probability of selections that are consistent with a particular language.

To achieve this goal, we propose a new approach for the integration of a language model and the EEG scores based on an N-th order Hidden Markov Model (HMM). [1] The thesis makes several contributions, which can be summarized as follows:

- The proposed approach presents the incorporation of an HMM-based language model into a P300-based spelling system.
- We demonstrate the use of our proposed approach on offline and online filtering and smoothing problems.
- We develop the first use of a Turkish language model within the context of BCI.

[1] A preliminary version of this work was published at The IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2013. [26].

Our approach has a number of features that differentiate it from previous work on language model-based BCI. These include the followings:

- Unlike the method presented in [15], our approach is fully probabilistic. It acknowledges that previous decisions contain uncertainties and performs prediction by taking into account the computed probabilities of all letters in the previous instant(s), rather than just the declared ones.

- Unlike the method presented in [16], our model takes advantage of both the past and the future. In this way, previously declared letters can be updated as new information arrives. Hence, error made in previous time could in principle be corrected at later time stages.

1.3 Thesis Outline

This thesis is organized as six chapters including the Introduction chapter.

- **Chapter 2** introduces the necessary background information about BCI, the P300 speller paradigm, the stimulus software used in our work and widely used classification techniques.

- **Chapter 3** presents all the technical pieces involved in the proposed language model based BCI system together with their mathematical preliminaries.

- **Chapter 4** presents the offline experiments we have conducted with subjects. In particular, the offline analysis method for the P300 speller, performance metrics and results of our experiments can be found in this chapter.

- **Chapter 5** presents in detail the methodology that was followed in our online experiments, including descriptions of the classification methods and decision making algorithms. The overall performance of our approach on multiple subjects is also reported.

- **Chapter 6** summarizes our work and presents a compilation of the results. A concluding discussion, and propositions for extensions and potential future work

directions in the scope of this thesis are also presented.

Chapter 2

Background on BCI and P300 Spellers

This chapter aims to provide the readers basic concepts about brain-computer interfaces (BCI), EEG signal processing, the P300 component of event-related potentials, and P300 spellers. A survey of published work, methods and results are also presented.

2.1 Introduction

A brain-computer interface (BCI) is a system that establishes a direct pathway between the brain and external devices. The BCI system has greatly assisted patients who suffered from some diseases in which all voluntary muscular control are lost such as Amyotrophic lateral sclerosis (ALS), brain stroke, and other neurological conditions, whose brains activity was impaired [11]. BCI serves as a bridge by connecting the brain and an external device. By using BCI technology, a user can directly communicate with or control external devices via the brain signals. The neural link between the brain and the computer is composed of two important components [27]. The first component is the interface to the brain that is responsible for the acquisition of the brain signals. The other component is on the computer and translates the brain signals into appropriate actions to interpret the user's intent. Both have been extensively studied in the past.

Acquisition of the brain signals can be done in several different ways. In terms of signal acquisition BCI procedures can be divided into invasive and non-invasive approaches. In invasive BCIs, the brain-activity is measured by getting as close as possible to the source of the brain signals. This method is widely used in early applications of the clinical diagnosis to track neurological disorders. Invasive BCIs are implanted directly into the

grey matter of the brain during neurosurgery. The advantage of this method is that it produces the highest quality signals since they lie in the grey matter and the signal is not interfered by cranial tissue [7]. However, invasive BCIs are affected by a build up of scar tissue around the electrodes. The drawbacks of invasive BCI are of course the surgery, but also the immense cost and the possible risk of infections [27]. However, with better understanding of brain waves as well as improvements in the techniques to measure brain activity, it is feasible to capture the signals without the need of surgery. This method is called non-invasive BCI. As it was mentioned in Chapter 1, electroencephalography (EEG), magnetoencephalography (MEG), positron emission tomography (PET), functional magnetic resonance imaging (fMRI) and functional near infrared spectroscopy (fNIRS) are non-invasive signal acquisition techniques. For practical BCI applications, a fast, portable and user friendly method is required so that patients can effectively use it. However, MEG, PET, and fMRI are technically demanding, expensive and hard to utilize outside a laboratory. Furthermore, PET, fMRI, and optical imaging, which depend on blood ow, have long time constants and thus are less amenable to rapid communication [7]. In contrary, EEG can function in most environments, require relatively simple and cheap equipments, and offer a new non-muscular communication and control channel [7].

2.2 Electroencephalography (EEG)

EEG is the most commonly used non-invasive BCI signal acquisition tool, mainly due to its good temporal resolution (in milliseconds), ease of use, portability and low set-up cost. EEG has been mainly used for clinical diagnosis of neurological disorders. It measures the electrical activity through the scalp via the electrodes attached to it. Although this is the most used technique, it has some serious disadvantages. It has poor spatial resolution, high noise levels and is more sensitive to activity in superficial layers of the cortex (i.e., activity deeper in the cortex will contribute less to the EEG signal) [27].

The working principle of EEG can be described as follows: First, the electrodes, placed on the scalp, are used to detect the EEG signals. Then, a connection of electrodes is used for amplifiers to magnify the EEG signals. Finally, a recording device can record the actual brain signals.

2.2.1 Electrodes

Electrodes, little flat pads of Ag/AgCl, are attached to the scalp with the help of an elastic cap. An example of the cap can be seen in Figure 2.1. A conductive gel is generally applied to the skin after abrasive skin preparation in order to decrease skin resistance or voltage offset and to have a stable, stationary conductive medium for proper measurements. However, electromagnetic interference, noise and signal degradation, need for skin preparation, etc., are problems for practical usage of these electrodes outside the laboratory [28]. Fortunately, to decrease the effect of the problems associated with high

Figure 2.1: 64-channel electrode cap using international 10-20 system for electrode distribution. Taken from [29].

electrode impedances and cable shielding, active electrodes such as those shown in Figure 2.2 have been developed. Active electrodes have very low output impedance and offer long term DC stability, which alleviates problems with regards to capacitive coupling between the cable and sources of interference, as well as any artefacts caused by cable and connector movements [30]. The electrodes are placed on the scalp of the subject according to an international system called the 10-20 system, proposed by the American EEG society [31]. This system recommends that the electrodes are placed in a 10%-20% distance from each other with respect to the total distance between the nasion and inion of the subject. The layout of the 64 channel EEG system that we use in our own recordings is presented in

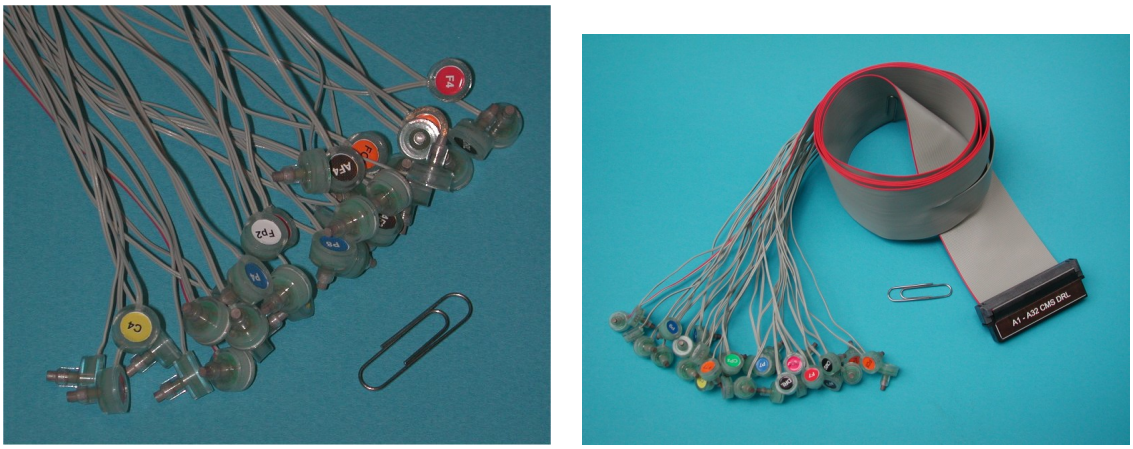


Figure 2.2: Active electrode sets used in this study. Taken from [32].

Figure 2.3.

2.3 A General BCI System

When a person is occupied with activities such as thinking, moving, feeling something, or he/she is stimulated by the external environment, the neurons in the brain are are also at work such that the brain will elicit electrical signals which contain physiological and pathological information [33]. Those electrical signals can be measured and acquired by a bio-signal acquisition system and further interpreted by a computer algorithm. By analysis and processing of these electrical signals, the brain activity is translated into command signals using a computer program, thus enabling the control of external devices [33]. A typical BCI system first records the brain activity and then translates it into control commands in order to control devices such as computers, electrical appliances as well as robots. A typical BCI system usually involves three parts as shown in Figure 2.4 : Signal Acquisition, Signal Processing and Application Interface.

The Signal Acquisition part acquires and amplifies brain signals. It uses (active) electrodes and an EEG amplifier. The Signal Processing part then processes the acquired brain signals in three steps sequentially: data preprocessing, feature extraction and classification. The processed data is transmitted into the Application Interface part for further control of external devices [33]. For controlling of the external devices, current studies

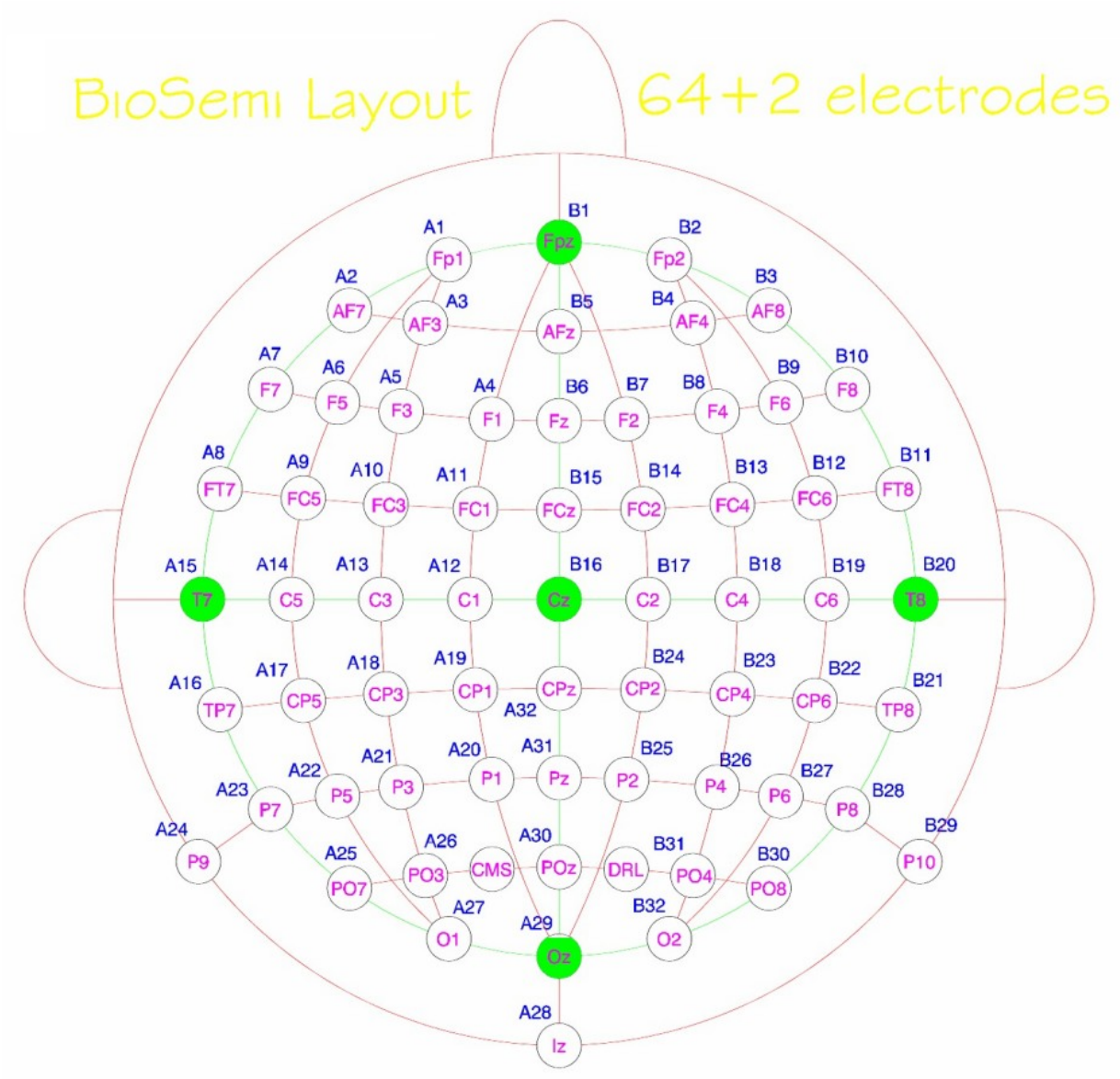


Figure 2.3: Electrode placement layout according to 10-20 electrode system. Taken from [29].

on BCI allow the users to perform several actions such as controlling robot arms, typing letter on a computer screen, moving a cursor, controlling a prostheses for various tasks such as a motorized wheel chair, etc [28].

Several well-known neural mechanisms are used considered in BCI applications. The most widely used ones include motor imagery [8], event-related potentials, steady-state visually evoked potentials [35], and slow cortical potentials [36]. Here we review the first

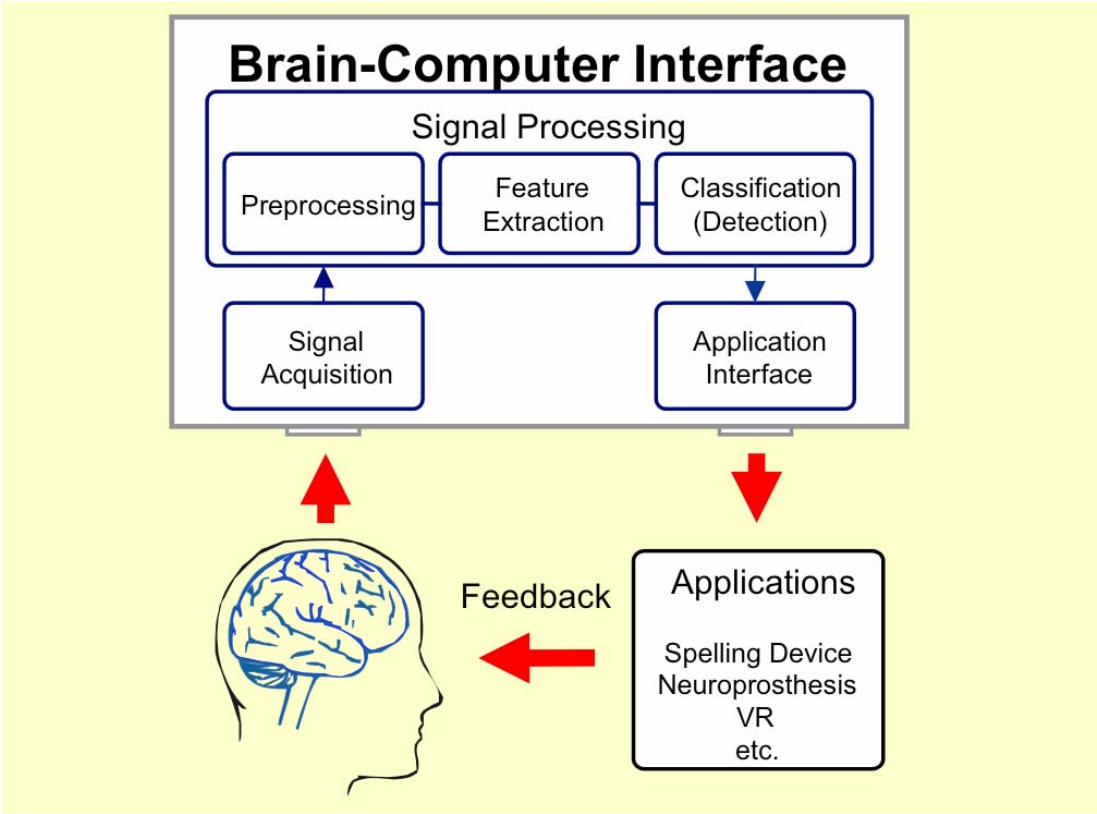


Figure 2.4: Illustration of a typical BCI system. Taken from [34].

two mechanisms.

2.3.1 Motor Imagery

Motor Imagery is one of the most popular BCI tasks that require the subjects to mentally imagine or simulate a physical action. Using EEG it is possible to record the brainwaves during that mental state. The EEG signal are recorded multiple times while the brain processes. The information is averaged over the different recordings to filter out redundant brain activity and to keep the relevant information [27]. Commonly data belonging to two classes such as mentally thinking about right and left hand movement are recorded [37]. This enables subjects to communicate choices between two categories by just thinking of movement of the right or left hand. A training session is needed to train the computer to differentiate between the different classes.

2.3.2 Event Related Potentials

Event-related potential (ERP) is any scalp recorded electro-physiological response that is the direct result of a thought or of a perception to an internal or external stimulus [38]. ERPs can be measured before, during or after a sensory, motor or psychological event [39, 40] and usually have a fixed time delay after (or before) the event, named stimulus. The ERPs are characterized by the time after the stimuli and a positive or negative large deflection of the signal. As in motor imagery there is no need to train the subject but a training session is needed for the computer to learn the particular ERP features of the individual. In case of ERPs it is even not required for persons to undertake particular actions, because the ERP is elicited involuntarily. One of the most extensively used ERP component in BCI research is P300 component. This thesis is completely related P300 component as well. In the next section, this component will be mentioned.

P300 component

The P300 is a type of Event-related potential (ERP) which is elicited by infrequent, task-relevant stimuli. It is the most widely studied ERP component. It usually appears as a large positive deflection in voltage which occurs at around 300ms to 600ms after the target stimulus onset [41]. The P300 signal is considered to be an endogenous potential because it occurs not because of the physical attributes of stimulus but the reaction of the subject [33]. The P300 component, usually named P3, appears around 300 ms with a positive voltage after the stimulus. This idea elicited another paradigm known as the 'oddball paradigm', where the subject is stimulated with two categories of events - relevant and irrelevant [42]. The relevant events occur rarely with respect to irrelevant events, and due to the complete random order of events, elicit a large P300 response in ERPs. In 1988, Farwell and Donchin used this paradigm to develop a communication system where subjects were able to type letters on a computer screen only by thought - with P300 signals [13]. Farwell and Donchin present a 6x6 matrix of letters and numbers to the subject. The rows and columns of the matrix are intensified in a block-randomized fashion, and the user is required to mentally count the number of occurrences of a target stimulus that contains the target letter. Here, the row and column that contain the target letter are the relevant

events or target stimuli, where in a block of 12 flashes, there are two such events. The other events, rows and columns that do not include the target letter are the irrelevant events or non-target stimuli, and there are ten such events in a block consisting 12 flashes [28].

The P300 component has a wide distribution along the mid-line scalp sites. Central-parietal (Cz) and mid-frontal (Fz) location are basically known to have highest amplitudes of the P300 component [33]. Figure 2.5 shows a typical P300 response averaged over trials recorded at electrode site Cz.

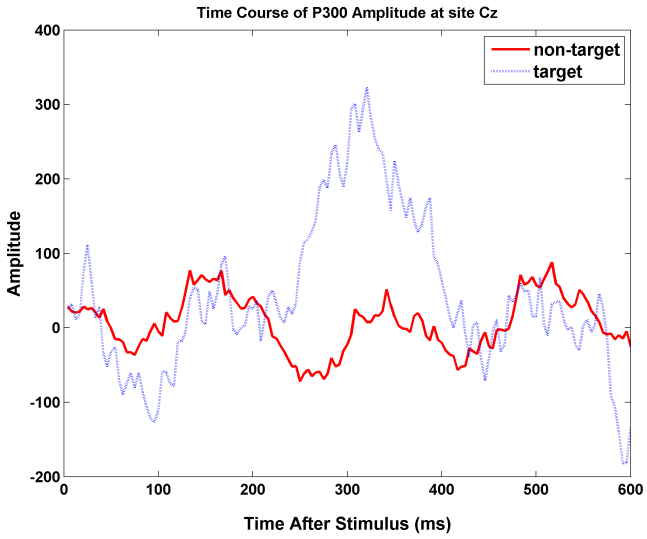


Figure 2.5: Average of brain signals over trials following a visual stimuli obtained from the central zero (Cz) electrode. The blue dashed line is the average response of trials where a P300 wave is visible, the solid red line shows the average response of trials where no P300 wave is elicited

Various factors determine the quality of the recorded P300 signals, as follows [33]:

- A subject's mental state, emotion, psychological activities, degree of fatigue and concentration will all effect the result of P300 recordings.
- The position of the electrodes and references should be carefully selected for obtaining P300 signals with best quality.

- The data processing procedure of recorded EEG data will also influence the fnal acquisition of P300 signal. Noise in the raw EEG data should be reduced in such a way to give the most undistorted P300 signal. P300 signal is always averaged by several measurements due to its small amplitude (in μv).

2.4 P300 based BCI systems

The P300-based BCI system has been widely studied since its first development in 1988. Recently, P300 based BCI systems and related technologies have been highly developed and improved. Donchin's first P300 speller [13] has become the most widely studied P300 based BCI system. Figure 2.6 shows the prototype of the first P300 speller paradigm [13]. Here, the task is to spell the word "B-R-A-I-N" letter by letter using the paradigm shown in the figure. The paradigm is a 6×6 matrix made up of 36 cells. It involves 26 letters of the alphabet and several other commands and symbols. The subject is asked to focus his/her gaze on the character that he/she wants to spell while each row and column of the matrix is flashed. The row and column flashes are in a random order. Whenever the desired character is intensified with either a row or a column, there will be a P300 component elicited at the stimulus onset [33]. With proper P300 feature selection and classification, the attended character of the matrix can be estimated and then displayed to the subject.

As opposed to the matrix layout of the popular P300 speller, new flashing paradigms and interactive forms have also been introduced. One example of this is the hexagonal two-level hierarchy of the Berlin BCI known as "Hex-o-Spell" [43] where multiple characters are displayed in an appealing visualization based on hexagons (see Figure 2.7 (a)). Another well established paradigm is the rapid serial visual presentation (RSVP) keyboard [44] in which visual stimulus sequences are displayed on a screen over time on a fixed focal area and in rapid succession (see Figure 2.7 (b)).

Another popular software tool BCI based spelling is BCI 2000 (see Figure 2.8 for a screenshot). BCI 2000 is a complete set of tools used by EEG research groups all over the world. It was first developed by the members of Schalk lab and presented in [45]. Featuring a module-based system, BCI 2000 has the capability of data acquisition from

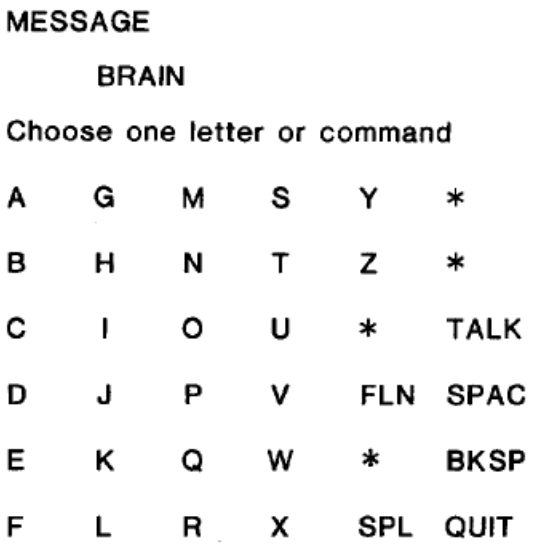
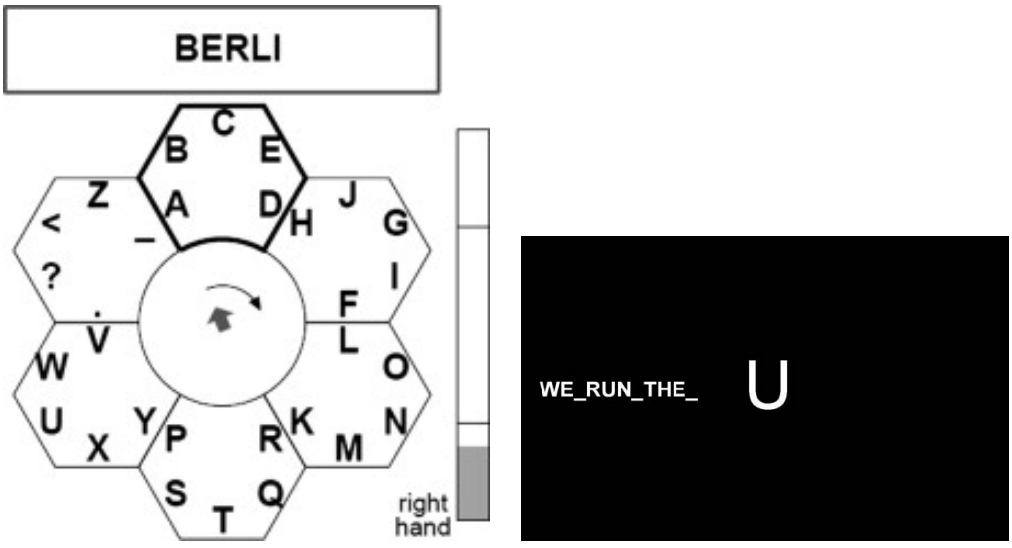


Figure 2.6: First P300 speller paradigm used by Donchin

several hardware, two stage (feature extraction and feature translation) signal processing phase, application interface where the subject decides an action with the help of translated control signals, and an operator interface to set various parameters and monitor other



(a) (b)

Figure 2.7: Two different flashing paradigms (a) Hex-o-Spell interface, (b) RSVP interface

software and/or experiment related information [28].

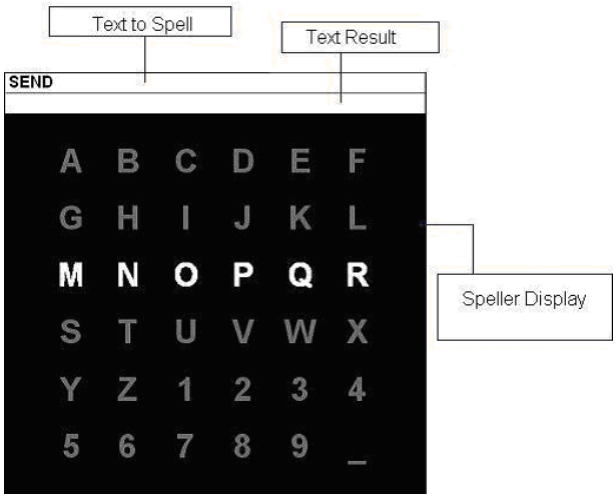


Figure 2.8: A screenshot of the BCI 2000 P300 speller application. *Text To Spell* indicates the pre-defined target letters. The speller will analyze evoked responses and will append the selected letter to *Text Result*. Taken from [46].

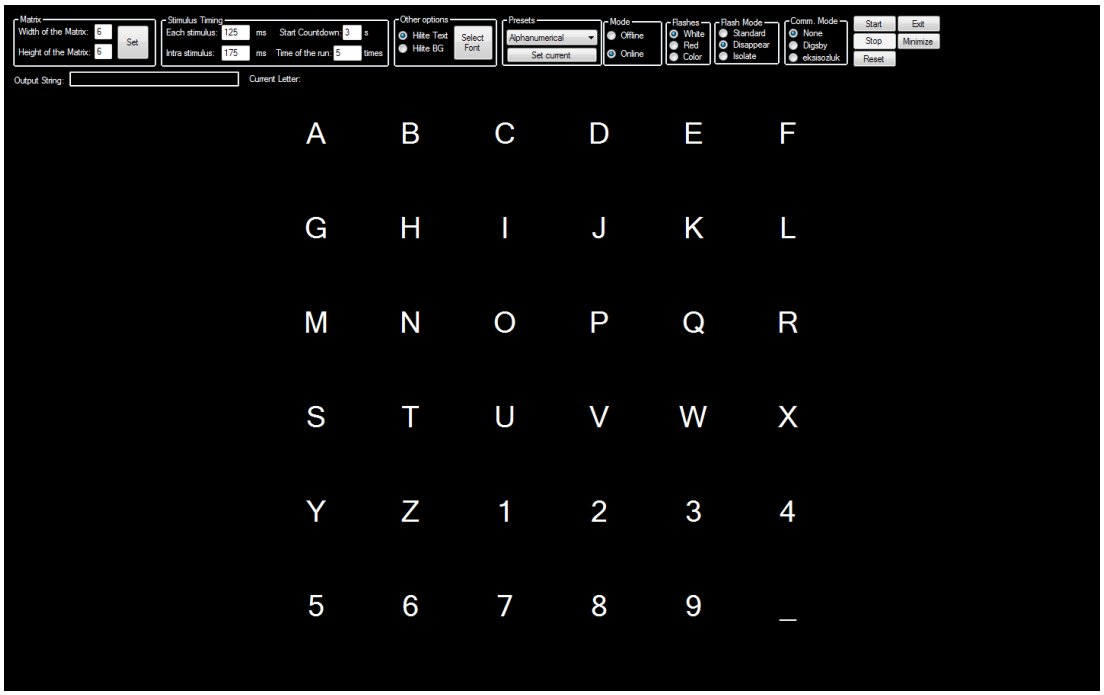


Figure 2.9: A screenshot of the SU-BCI P300 Speller before the beginning of the session.

In this study, the SU-BCI P300 stimulus software previously developed at the Signal Processing and Information Systems (SPIS) Laboratory [28] is used to deliver the subject the required visuals, or directions, to evoke the necessary potentials. It is essentially a matrix based system , first introduced by Donchin [13]. Since the SPIS Laboratory has plans for further studies in the P300 speller context, the software had to satisfy diverse needs. Therefore, the software architecture is built so that the broad needs of different P300 experiments can be satisfied within a single software by allowing the user to derive numerous analyses and cross-analyses within the context of a P300 speller [28]. A screenshot of the SU-BCI P300 stimulus software is presented in Figure 2.9.

2.4.1 Data processing procedure for a P300-based BCI system

The goal of data analysis is to identify the subject's P300 component from the detected EEG signal and extract those signals which reflect the characteristic parameters of the subject. The signals are then converted to executable commands to control external devices through appropriate algorithms [33].

In P300 based BCI system, the flashing of the rows and columns are used for the visual stimuli of the ERP. The random order is needed to make the row or column flashes unpredictable for the subject in order to comply with the need of a visual stimulus on an unexpected moment. Gazing is needed to make sure the P300 wave is only elicited if the column or row the subject focuses on is intensified. By correlating the timing of the occurrence of this wave with the intensified columns and rows, the focused letter can be determined [27]. For this reason, the problem is reduced to a binary classification problem of whether the short EEG data (epoch) includes the P300 wave or not [28].

The EEG data processing procedure consists of three steps: data pre-processing, classification and post processing. First the raw EEG is preprocessed for the preparation of classification. A digital filtering process is included in the first step where a band-pass filter is usually applied and the signals are decimated or sub-sampled by a factor to eliminate the artefacts. Then, the EEG data are split into epochs corresponding to individual row and column flashes. After the end of first step, a feature extraction process is needed to obtain a better representation of the data with different features. Features might be

peaks, actual or special waveforms or deflections at specific times, spectral density, etc. In the scope of this thesis, the features are almost an imitation of the actual waveform, in other words, the amplitude of the signals for that period [28]. The second step is classification process of the occurrence of P300 wave per column and row. This is done by giving the formed feature vector in the previous intermediate step to the classifier. For every EEG epoch data represented with a feature vector, the classifier returns a value corresponding to its similarity to the attended class containing a P300 signal. The last step, post-processing, takes the P300 detection results for every column and row and combines them to determine the corresponding letter which is ideally the letter at the intersection of row and column exhibiting P300 responses.

2.5 P300 Speller classification techniques

The EEG signals are classified based on different features generated from brain activities recorded at different electrode locations. The performance of signal classification depends on two factors: one is whether the signal being classified has a strong feature, the other is the effectiveness of the used classification algorithm [33]. Several type of classifiers have been practised before. This section gives some brief information about the classification and feature extraction approaches used in the P300 BCI context.

Fisher's Linear Discriminant Analysis (FLDA)

Fisher's linear discriminant analysis is a widely used classification method in the P300 speller. FLDA is a supervised classifier that intends to compute a discriminant vector that separates two or more classes as well as possible. FLDA tries to find a discriminant vector that results in data within a class get more concentrated and data between two classes (target and non-target) get more separated. The discriminant vector $\boldsymbol{w}$ is a function of these data, and the output of the analysis, given an input vector $\hat{\boldsymbol{x}}$, is simply $\boldsymbol{w}^{\mathrm{T}}\hat{\boldsymbol{x}}$ [47]. The best projection satisfies the following equation

$$\boldsymbol{w} = (\mathbf{S_1} + \mathbf{S_{-1}})^{-1}(\boldsymbol{m_1} - \boldsymbol{m_{-1}}) \tag{2.1}$$

where $\mathbf{S}$ and $\boldsymbol{m}$ represent the covariances and means of two classes ± 1 respectively, which need to be separated [33]. The output values obtained by FLDA can be used in this way: the maximum of the output values might be summed over multiple trials, and then the intersection of the row and column satisfying maximum value among all is chosen as the answer of the classification. A detailed description of FLDA is given in Appendix A of [47]. This method has been extensively practised in P300 studies (see, e.g., [48]).

Stepwise Linear Discriminant Analysis (SWLDA)

Stepwise Linear Discriminant Analysis (SWLDA) is a technique for selecting suitable predictor variables to be included in a multiple regression model that has proven successful for discriminating P300 Speller responses. A combination of forward and backward stepwise regression is implemented. Starting with no initial model terms, the most statistically significant predictor variable having a $p - value < 0.1$, is added to the model. After each new entry to the model, a backward stepwise regression is performed to remove the least significant variables, having $p - values > 0.15$. This process is repeated until the model includes a predetermined number of terms, or until no additional terms satisfy the entry/removal criteria [49]. This classification technique is applied in [19, 50] and results are reported.

Support Vector Machine (SVM)

SVM has become popular in machine learning and is considered as one of the most accurate classifiers in P300 speller research. The primary idea of SVM is to determine a separating hyperplane (see Figure 2.10) between two classes which can maximize the distance between the hyperplane and the closest points from both classes that constitute the support vectors [51]. In other words, the margin between classes needs to be maximized. However, since the samples of the classes in EEG settings are quite inseparable from each other due to the variability of background activity of brain signals, non-linear kernels should be applied instead of linear SVM kernels [28]. In [18, 52], different types of SVM were performed and the results were demonstrated.

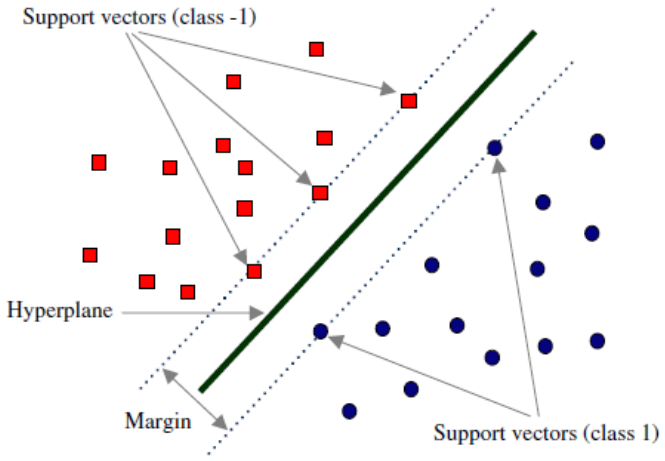


Figure 2.10: Optimal hyperplane for support vector machine with two classes. Taken from [53].

Bayesian Linear Discriminant Analysis (BLDA)

BLDA can be seen as an extension of Fisher's Linear Discriminant Analysis (FLDA). In contrast to FLDA, in BLDA regularization is used to prevent overfitting to high dimensional and possibly noisy datasets. Through a Bayesian analysis, the degree of regularization can be estimated automatically and quickly from training data without the need for time consuming cross-validation [54]. The mathematical preliminaries of BLDA will be presented in Chapter 3. In addition to this, BLDA has been widely practised in BCI settings such as in [55,56].

Independent Component Analysis (ICA)

Independent Component Analysis (ICA) is a type of blind source separation method that can break a mixed signal down to statistically independent components by maximizing their non-Gaussianity. The components are related to different features of the signal. One can map them and determine which ones are connected with P300. In other words, ICA has the ability to reveal the hidden features even if they are buried in the background noise. This ability makes it possible to detect P300 via a single trial [57]. ICA is successfully applied in EEG signal classification (see, e.g., [58]).

2.6 Language Model

A language model is a mathematical model of a particular natural language which characterizes, captures and exploits the rules defining that natural language [59]. Language modelling has many applications and has been extensively used in various areas such as Automatic Speech Recognition (ASR), machine translation, part-of-speech tagging, information retrieval, text input, etc.

We know that in a particular language, a word is not an arbitrary sequence of letters, in fact, it follows some rules inherent to the language and common uses. If only the beginning of a word is known, then, it is often possible to complete the word or at least predict the best possible words that would complete the word. The task of estimating a letter will be even easier when the preceding and succeeding letters are provided [27]. Given the context of a word, different letter sequences can be formed based on the context such that some sequences will occur more and others will occur less. A statistical language model tries to capture these probabilities by assigning a probability distribution over sequences of words or letters [27]. Since a P300 based BCI system is designed to provide a means for communication by enabling subjects to spell some text or meaningful letter sequences, i.e, words or sentences, the letter probabilities obtained from a statistical language model can be used as prior information in the decision algorithm for letter estimation [15]. Based on this observation, this thesis proposes to exploit a language model in conjunction with the information coming from EEG data to merge them in a single decision-making algorithm. The proposed model will be discussed with its mathematical preliminaries in Chapter 3. Chapter 1 already contains brief information about the relevant existing works that incorporate language models into the P300 speller setting. The comparison of our model with these relevant works will be provided at the end of Chapter 3.

2.7 Summary

This chapter provides a general discussion and background knowledge about the topics related with this thesis study. In particular, the concepts of EEG, Event-related potential (ERP), P300 component and P300-based BCI systems were described. The P300-based

BCI section mainly involves the information about the application of P300 speller, including the working principle of the P300 speller, different flashing interfaces being used in P300 context, and the data processing procedure that can be used to estimate the typed letter given the brain signals. Several commonly used classification techniques utilized in the context of P300 speller are also briefly mentioned. At last, a short discussion of statistical language modelling is provided and the motivation to use a language model in this study is discussed within this context.

Chapter 3

Language Model-based P300 Speller

In this chapter, we describe in detail the proposed classification algorithm based on a language model. The classification algorithm is composed of two steps [26]:

1. Either Bayesian Linear Discriminant Analysis (BLDA) or Logistic Regression (LR) 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 an n-gram language model, the classifier decides on each letter in the sequence by using either Forward, Forward-Backward, or Viterbi algorithms.

Figure 3.1 visualizes the system diagram of the proposed model incorporating of a language model to make a prediction on the target letter into P300 based speller. The following sections provide the details of the algorithm. This chapter focuses on the language model-based classification algorithm. The stimulus software used during EEG data acquisition and data pre-processing methods used in this study will be described in detail in Chapter 4.

3.1 Bayesian Linear Discriminant Analysis (BLDA)

For the first step of our classification process, one of the approaches we consider and apply is a type of linear classifier called BLDA. This section exactly follows Appendix B of [47] where a summary of BLDA is given. A more detailed explanation is provided in [54].

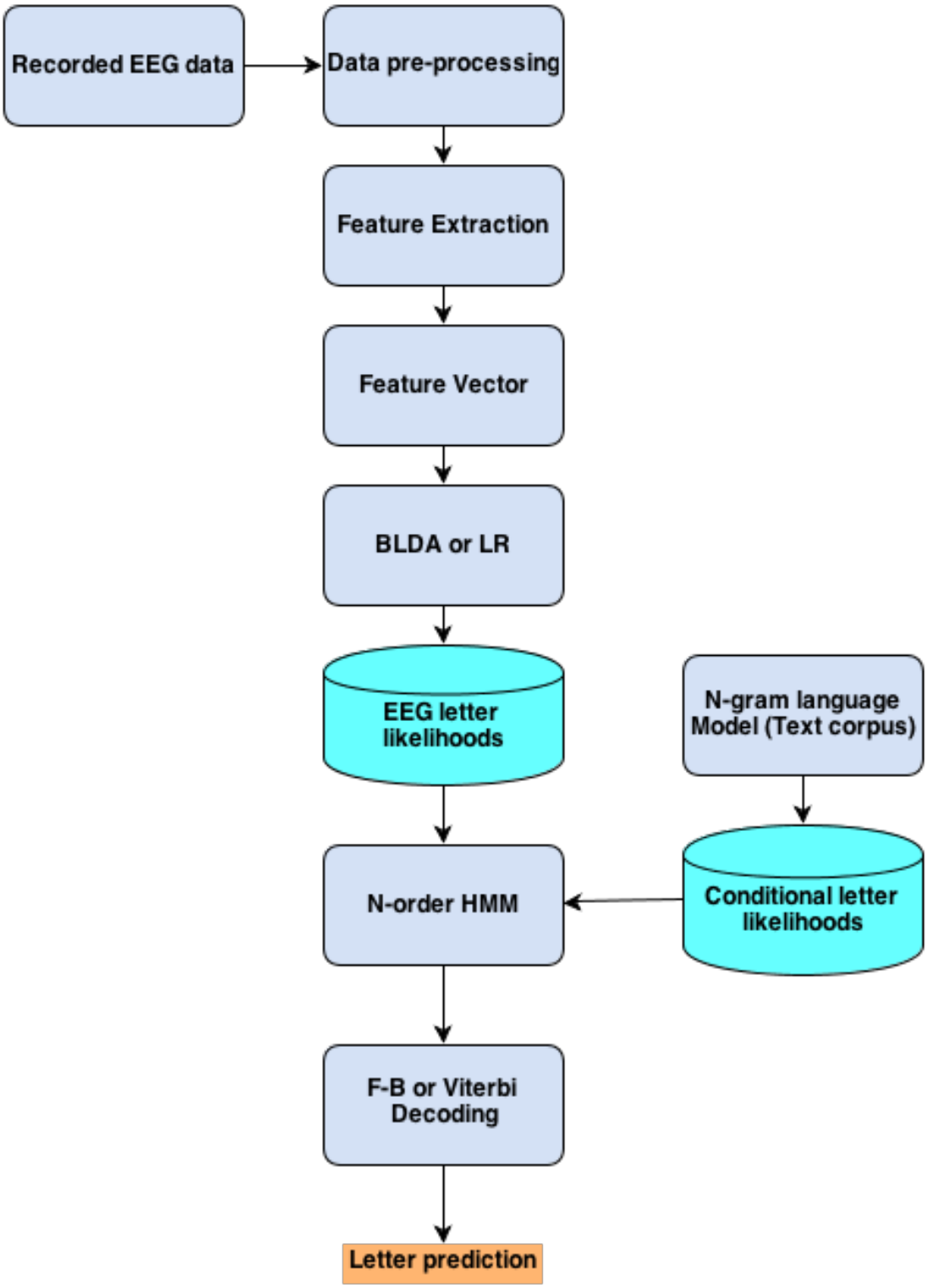


Figure 3.1: The system diagram of the proposed P300 recognition system in this study.

BLDA can be seen as an extension of Fisher's Linear Discriminant Analysis (FLDA). In contrast to FLDA, in BLDA regularization is used to prevent overfitting to high dimensional and possibly noisy datasets. Through a Bayesian analysis, the degree of regularization can be estimated automatically and quickly from training data without the need for the time consuming cross-validation process.

Least squares regression is equivalent to FLDA if regression targets are set to N/N_1 for examples from class 1 and to $-N/N_2$ for examples from class -1; where N is the total number of training examples, N_1 is the number of examples from class 1 and N_2 is the number of examples from class -1. Given the connection between regression and FLDA, BLDA performs regression in a Bayesian framework and sets the targets mentioned above.

The assumption in Bayesian regression is that targets t and feature vectors x are linearly related with additive white Gaussian noise n.

$$t = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + n \tag{3.1}$$

Given this assumption, the likelihood function for the weights $\boldsymbol{w}$ used in regression is

$$p(\mathbf{D}|\beta, \boldsymbol{w}) = \left(\frac{\beta}{2\pi}\right)^{N/2} \exp\left(-\frac{\beta}{2}\|\mathbf{X}^{\mathrm{T}}\boldsymbol{w} - \boldsymbol{t}\|^2\right) \tag{3.2}$$

Here, $\boldsymbol{t}$ denotes the vector containing the regression targets, $\mathbf{X}$ denotes the matrix that is obtained from the horizontal stacking of the training feature vectors, $\mathbf{D}$ denotes the pair $\{\mathbf{X}, t\}$, β denotes the inverse variance of the noise, and N denotes the number of examples in the training set.

To perform inference in a Bayesian setting , one has to specify a prior distribution for the latent variables, i.e., for the weight vector $\boldsymbol{w}$. The expression for the prior distribution we consider and use here is

$$p(\boldsymbol{w}|\alpha) = \left(\frac{\alpha}{2\pi}\right)^{D/2} \left(\frac{\epsilon}{2\pi}\right)^{1/2} \exp\left(-\frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\mathbf{I}'(\alpha)\boldsymbol{w}\right) \tag{3.3}$$

where $\mathbf{I}'(\alpha)$ is a square, $D+1$ dimensional, diagonal matrix

$$\mathbf{I}'(\alpha) = \begin{bmatrix} \alpha & 0 & \dots & 0 \\ 0 & \alpha & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \epsilon \end{bmatrix}$$

and D is the number of features. Hence, the prior for the weights is an isotropic, zero-mean Gaussian distribution. The effect of using a zero-mean Gaussian prior for the weights is similar to the effect of regularization term used in ridge regression and regularized FLDA. The estimates for $\boldsymbol{w}$ are shrunk towards the origin and the danger of over-fitting is reduced. The prior for the bias (the last entry in $\boldsymbol{w}$) is a zero-mean univariate Gaussian. Setting ϵ to a very small value, the prior for the bias is practically flat. This expresses the fact that a priori there are no assumptions made about the value of the bias parameter.

Given the likelihood and the prior, the posterior distribution can be computed using Bayes rule.

$$p(\boldsymbol{w}|\beta, \alpha, \mathbf{D}) = \frac{p(\mathbf{D}|\beta, \boldsymbol{w})p(\boldsymbol{w}|\alpha)}{\int p(\mathbf{D}|\beta, \boldsymbol{w})p(\boldsymbol{w}|\alpha)\mathrm{d}\boldsymbol{w}} \tag{3.4}$$

Since both the prior and the likelihood are Gaussian, the posterior is also Gaussian and its parameters can be derived from the likelihood and the prior by completing the square. The mean $\boldsymbol{m}$ and covariance $\mathbf{C}$ of the posterior satisfy the following equations.

$$\boldsymbol{m} = \beta(\beta\mathbf{X}\mathbf{X}^{\mathrm{T}} + \mathbf{I}'(\alpha))^{-1}\mathbf{X}t \tag{3.5}$$

$$\mathbf{C} = (\beta\mathbf{X}\mathbf{X}^{\mathrm{T}} + \mathbf{I}'(\alpha))^{-1} \tag{3.6}$$

By multiplying the likelihood function Eq. (3.2) for a new input vector $\hat{\boldsymbol{x}}$ with the posterior distribution Eq.(3.4) followed by integration over $\boldsymbol{w}$, we obtain the predictive distribution, i.e., the probability distribution over regression targets conditioned on an input vector,

$$p(\hat{t}|\beta, \alpha, \hat{\boldsymbol{x}}, \mathbf{D}) = \int p(\hat{t}|\beta, \hat{\boldsymbol{x}}, \boldsymbol{w})p(\boldsymbol{w}|\beta, \alpha, \mathbf{D})\mathrm{d}\boldsymbol{w} \tag{3.7}$$

The predictive distribution is Gaussian and can be characterized by its mean μ and its variance σ^2.

$$\mu = \boldsymbol{m}^{\mathrm{T}}\hat{\boldsymbol{x}} \tag{3.8}$$

$$\sigma^2 = \frac{1}{\beta} + \hat{\boldsymbol{x}}^{\mathrm{T}}\mathbf{C}\hat{\boldsymbol{x}} \tag{3.9}$$

In this study, we only use the mean value of the predictive distribution for taking decisions. The classification problem in our setting 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, i.e, mean value of the predictive distribution, 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 [26].

In a more general setting, class probabilities could be obtained by computing the probability of the tar- get values used during training. Using the predictive distribution from Eq. (3.7) and omitting the conditioning on β,α, $\mathbf{D}$, we obtain

$$p(\hat{y} = 1|\hat{\boldsymbol{x}}) = \frac{p(\hat{t} = \frac{N_1}{N}|\hat{\boldsymbol{x}})}{p(\hat{t} = \frac{N_1}{N}|\hat{\boldsymbol{x}}) + p(\hat{t} = \frac{-N_2}{N}|\hat{\boldsymbol{x}})} \tag{3.10}$$

Both the posterior distribution and the predictive distribution depend on the hyperparameters α and β. We have assumed above that the hyperparameters are known, however in real-world situations the hyperparameters are usually unknown. One possibility to solve this problem would be to use cross-validation to determine the hyperparameters that yield the best prediction performance. However, the Bayesian regression framework offers a more elegant and less time-consuming solution for the problem of choosing the hyperparameters. The idea is to write down the likelihood function for the hyperparameters and then maximize the likelihood with respect to the hyperparameters. The maximum likelihood solution for the hyperparameters can be found with a simple iterative algorithm [51].

3.2 Logistic Regression

As an alternative to BLDA, the second approach we consider and use for the first step of our classification process is logistic regression (LR). LR is based on a discriminative

training model, and is performed to directly model the posterior probabilities of the classes (P300 versus not) given the EEG data. The rest of this section mainly follows the detailed explanations for LR in [60].

Logistic Regression is an approach for learning functions of the form $f : X \rightarrow c$, or $P(c|X)$ in the case where c is discrete-valued and $X =< X_1, X_2, ...X_n >$ is any vector containing discrete or continuous variables. Logistic regression assumes a parametric model for the distribution $P(c|X)$, then directly estimates its parameters from the training data. Moreover, it models the posterior probabilities of the classes by a generalized linear model while at the same time the sum of two probabilities must equal to 1 and remain in [0,1]. The parametric models are as follows:

$$P(c = 1|X) = \frac{1}{1 + \exp(w_0 + \sum_{j=1}^{n} w_j X_j)} \tag{3.11}$$

$$P(c = -1|X) = \frac{\exp(w_0 + \sum_{j=1}^{n} w_j X_j)}{1 + \exp(w_0 + \sum_{j=1}^{n} w_j X_j)} \tag{3.12}$$

Here, in our model, X represents the EEG data feature vector corresponding to a flash or epoch, $c = 1$ represents the attended class and $c = -1$ represents the non-attended class.

3.2.1 Estimating Logistic Regression Parameters

Suppose that we have a training set of i.i.d. samples $\mathcal{D} = (c^{(l)}, X^{(l)})_{l=1}^{M}$ drawn from a training distribution. A reasonable approach for training Logistic Regression is to find the parameter values maximizing the conditional data likelihood. The estimated parameters W satisfy

$$W \leftarrow \arg\max_{W} \prod_{l} P(c^l|X^l, W) \tag{3.13}$$

where $W =< w_0, w_1, ...w_n >$ is the vector of parameters to be estimated, c^l denotes the observed class label value of c in the lth training example and X^l denotes the EEG data for the lth flash stimulus in the stimulus sequence X of the training data. If we take the

log of the conditional likelihood, we obtain:

$$W \leftarrow \arg\max_{W} \sum_{l} \ln P(c^l | X^l, W) \tag{3.14}$$

This conditional data log likelihood can be written as

$$L(W) = \sum_{l} c^l \ln P(c^l = 1 | X^l, W) + (1 - c^l) \ln P(c^l = -1 | X^l, W) \tag{3.15}$$

By using the flipped version of the assignment of c in Eq.(3.11) and Eq.(3.12), we can re-define the log of the conditional likelihood as

$$\begin{aligned} L(W) &= \sum_{l} c^l \ln P(c^l = 1 | X^l, W) + (1 - c^l) \ln P(c^l = -1 | X^l, W) && (3.16) \\ &= \sum_{l} c^l \ln \frac{P(c^l = 1 | X^l, W)}{P(c^l = -1 | X^l, W)} + \ln P(c^l = -1 | X^l, W) && (3.17) \\ &= \sum_{l} c^l (w_0 + \sum_{j}^{n} w_j X_j^l) - \ln(1 + \exp(w_0 + \sum_{j}^{n} w_j X_j^l)) && (3.18) \end{aligned}$$

where X_j^l denotes the value of X_j for the lth training example.

Unfortunately, there is no closed form solution to maximizing $L(W)$ with respect to W. One commonly used approach is to use gradient ascent, in which we can make use of gradient information of the likelihood , and then ascend the likelihood. The ith component of the gradient vector has the form

$$\frac{\partial L(W)}{\partial w_j} = \sum_{l} X_j^l (c^l - \hat{P}(c^l = 1 | X^l, W)) \tag{3.19}$$

where $\hat{P}(c^l | X^l, W)$ is the predicted conditional likelihood value using Eq.(3.11-3.12) and the weight vector W. To accommodate weight w_0, we assume an illusory $X_0 = 1$ for all l.

Given this formula for the derivative of each w_j, we can use standard gradient ascent to optimize the weights W. Beginning with initial weights of zero, we iteratively update the weights in the direction of the gradient, on each iteration changing every weight w_j according to following relation:

$$w_j \leftarrow w_j + \eta \sum_{l} X_j^l (c^l - \hat{P}(c^l = 1 | X^l, W)) \tag{3.20}$$

where η is the *learning rate* chosen as a small constant (e.g., 0.1) to ensure convergence of the method. Since $L(W)$ is concave, this gradient ascent procedure will converge to

a global maximum. A more detailed explanation about gradient ascent/descent can be found in [45].

3.2.2 Regularization in Logistic Regression

Overfitting the training data is a problem that can occur in Logistic Regression, especially when the data are very high dimensional and training data are sparse. One approach to reduce overfitting is regularization, in which we create a modified penalized log likelihood function which penalizes large values of W. Then, the penalized log likelihood function becomes

$$W \leftarrow \arg\max_{W} \sum_{l} \ln P(c^l|X^l, W) - \frac{\lambda}{2}\|W\|^2 \tag{3.21}$$

which adds a penalty proportional to the square magnitude of W. λ is the constant regularization parameter.

Modifying the objective by adding in this penalty term gives us a new objective to maximize. It is easy to show that maximizing it corresponds to calculating a MAP estimate for the parameter W if we assume that the prior distribution $P(W)$ is a normal distribution with mean zero, and a variance related to $1/\lambda$. Note that, the MAP estimate for W involves optimizing the objective

$$\sum_{l} \ln P(c^l|X^l, W) + \ln P(W) \tag{3.22}$$

Here, if $P(W)$ is a zero mean Gaussian, then $\ln P(W)$ yields a term proportional to $\|W\|^2$.

Given the penalized log likelihood function, the derivative of this penalized log likelihood is similar to earlier derivative in Eq.(3.19) with one additional term. The modified gradient descent rule becomes

$$w_i \leftarrow w_i + \eta \sum_{l} X_j^l (c^l - \hat{P}(c^l = 1|X^l, W) - \eta\lambda w_i \tag{3.23}$$

To obtain a λ value for each subject, we choose 10 different λ values in the interval [0,10] and apply leave-one-out cross-validation within the training data of the each subject to decide on which λ to use.

3.3 Language Model-based BCI

We believe that combining the letter likelihood probability scores obtained by either BLDA or Logistic Regression 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 EEG data corresponding to a run (all trial groups for typing a character) form the observed variable [26]. Note that we do not perform HMM training within this model. Instead, we perform training separately and learn the necessary HMM parameters using supervised classifiers and a text corpus (for detailed explanation see Section 3.3.1). A diagram illustrating a sequential chain of an HMM is represented in Figure 3.2.

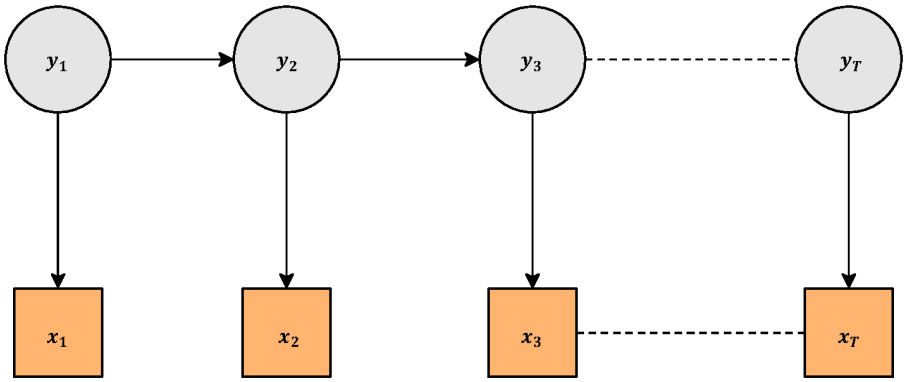


Figure 3.2: A sequential HMM

In our model, $Y = \{y_1, y_2, ..., y_T\}$, where each $y_i = j \in S$, S is the set containing all elements in the speller matrix and $X = \{x_1, x_2, ..., x_T\}$, each x_i represents the EEG scores of all symbols in the matrix corresponding to the time instant i, hence x_i is a 36 -dimensional vector. For an N-th order HMM, the conditional distribution of Y given X is proportional to the joint probability:

$$p(Y|X) \propto p(y_1) \prod_{i=1}^{N} p(y_i|y_{i-1}) p(x_i|y_i) \prod_{i=N+1}^{T} p(y_i|y_{i-N}, ..., y_{i-1}) p(x_i|y_i) \tag{3.24}$$

We will describe in Section 3.3.1 how to obtain the emission, $p(x_i|y_i)$, and transition probabilities, $p(y_i|y_{i-N}, ..., y_{i-1})$, stated in Eq.(3.24). To make an inference on this HMM,

we can choose to maximize either the marginal or joint probabilities of a letter sequence given the EEG data. For all i in $[1, T]$,

$$y_i' = \arg\max_j p(y_i = j|x_{1:T}) \tag{3.25}$$

$$Y' = \arg\max_Y p(Y|x_{1:T}) \tag{3.26}$$

Eq.(3.25) tries to estimate individually the most likely character for each time instant and can be efficiently solved by the Forward-Backward algorithm. Eq.(3.26) finds the most likely letter sequence given the model and can be solved by the Viterbi algorithm [61]. Note that both of these approaches use the entire data (both in the past and in the future) to estimate the letter at a particular time instant. Hence these are "smoothing" recursive estimation algorithms. One might also be interested in an algorithm that outputs the most likely letter based on data up to that time point. This could be done by the Forward algorithm which is a "filtering" approach that can operate in real time. Since the Forward algorithm is a straightforward special case of the Forward-Backward algorithm, we do not cover it in detail here, however in future chapters we present results based on the Forward algorithm as well.

3.3.1 Forward-Backward Algorithm

Let y_t denote the state at time t where $t \in \{1, 2, ..., T\}$ and let us define an observation sequence, $X = x_1 x_2 .... x_T$, where each x_k represents the EEG scores of all possible symbols for kth 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 defines the joint probability of the partial observation sequence until time t, (i.e., $x_{1:t}$) and the state at time t (i.e., y_t). In the second step, the algorithm computes backward probabilities providing the probability of the partial observation sequence from $t + 1$ to T, given the state i at time t. Then, we can combine these two sets of probabilities to estimate the probability distribution over states at any particular time as follows [62]:

$$P(y_t = i|x_{1:T}) \propto P(x_{1:t}, y_t = i)P(x_{t+1:T}|y_t = i) \tag{3.27}$$

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 $\beta_t(i)$.

For an N-th order HMM, $\alpha_t(i)$ and $\beta_t(i)$ can be recursively computed as follows:

$$\alpha_1(i_1) = P(y_1 = i_1)P(x_1|y_1 = i_1) \tag{3.28}$$

$$\alpha_t(i_t) = [\sum_{i_{t-N}} \dots \sum_{i_{t-1}} \alpha_{t-1}(i_{t-N}, ..., i_{t-2}, i_{t-1}) a_{i_{t-N},...,i_{t-1},i_t}] P(x_t|y_t = i_t) \tag{3.29}$$

where $a_{i_{t-N},...,i_{t-1},i_t} = P(y_t = i_t|y_{t-N} = i_{t-N}, ..., y_{t-1} = i_{t-1})$, $1 < t \leq T$ and each $i_{t-N}, ..., i_{t-1}, i_t \in S$. In a similar way, the backward probabilities are calculated as follows:

$$\beta_T(i_T) = 1 \tag{3.30}$$

$$\beta_t(i_t) = [\sum_{i_{t+1}} \dots \sum_{i_{t+N}} \beta_{t+1}(i_{t+1}, i_{t+2}, ..., i_{t+N}) a_{i_t,i_{t+1},...,i_{t+N}} P(x_{t+1}|y_{t+1} = i_{t+N})] \tag{3.31}$$

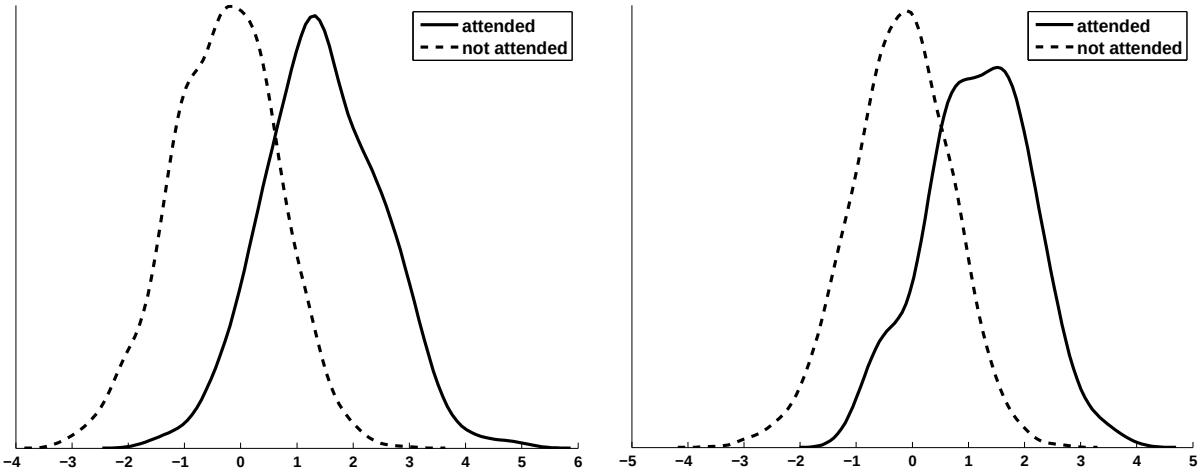
for $T - 1 \geq t \geq 1$.

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

$$P(x_t|y_t = i) = \prod_{n=1}^{N_t} p(x_t(i, n)|c_i = 1)(\prod_{n=1}^{N_t} \prod_{i' \in S \setminus \{i\}} p(x_t(i', n)|c_{i'} = -1)) \tag{3.32}$$

where $x_t(i, n)$ represents the epoch scores containing the character i at the nth trial group (repetition) and $x_t(i', n)$ represents those not containing the character i. Given the class label c_i of the flashes corresponding to letter $i \in S$, we have observed that $p(x_t(i, n)|c_i = 1)$ and $p(x_t(i', n)|c_{i'} = -1)$ are normally distributed by analyzing the distribution of the training data scores as shown in Figure 3.3. Test data epoch scores obtained by BLDA are converted to probability values by using density estimation of a Gaussian [63] whose parameters are estimated from training data scores for both attended and non-attended classes.

$$p(x_t(i, n)|c_i) = \begin{cases} \frac{1}{\sqrt{2\pi\sigma_a^2}} \exp\left(\frac{1}{2\sigma_a^2}(x_t(i, n) - \mu_a)^2\right) & \text{if } c_i = 1 \\ \frac{1}{\sqrt{2\pi\sigma_{na}^2}} \exp\left(\frac{1}{2\sigma_{na}^2}(x_t(i, n) - \mu_{na})^2\right) & \text{if } c_i = -1 \end{cases} \tag{3.33}$$



(a) subject 2 (b) subject 7

Figure 3.3: BLDA score distributions: histograms of the attended (solid curve) and non-attended (broken curve) scores from BLDA.

where μ_a, σ_a^2, μ_{na} and σ_{na}^2 are the means and variances of the distributions for the attended and non-attended flashes, respectively. The posterior probabilities obtained by applying Logistic regression to the test data can be turned into the following conditional probabilities by using Bayes'rule: $p(x_t(i,n)|c_i) = \frac{p(c_i|x_t(i,n))p(x_t(i,n))}{p(c_i)}$. Since $p(x_t(i,n))$ is independent of the label c_i, it can be discarded. Moreover, it is also clear that we can neglect $p(c_i)$ in the decoding since $\prod_{n=1}^{N_t} \frac{1}{p(c_i)}$ is the same constant for all $i \in S$. Note that by directly learning the conditional probabilities $p(c_i|x_t(i,n))$ through Logistic regression from the data, we perform a discriminative training for a generative model.

The initial probability, $\pi_i = P(y_1 = i)$ and the transition probabilities $a_{i_{t-n_{lm}+1},\ldots,i_{t-1},i_t}$ are estimated using an n-gram language model (n_{lm} is the order of the language model). N-grams for the Turkish language were obtained from a translation of a book which contains nearly 300,000 words including various types of lexicon. The conditional letter probabilities are obtained by using Katz Back-off smoothing technique that will be described in Section 3.3.3.

Having calculated the forward and backward probabilities based on Eq.(3.28-31), the probability of being in state i at time t given the all observation sequence X can be

expressed as follows [61]:

$$P(y_t = i|x_{1:T}) = \gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_i \alpha_t(i)\beta_t(i)} \tag{3.34}$$

We can estimate the individually most likely state or character at any time t as follows:

$$\hat{y_t} = \arg\max_i[\gamma_t(i)], \qquad 1 \leq t \leq T. \tag{3.35}$$

3.3.2 Viterbi Algorithm

The forward-backward algorithm can be used to determine the most likely character for any kth letter of a target word. However, it can not find the most likely letter sequence for a given model. In order to find the single best letter sequence, we use the Viterbi algorithm on our proposed HMM [64, 65]. The required state transition probabilities and observation symbol probabilities for this algorithm were already provided in Section 3.3.1. For each trial group, the Viterbi algorithm produces the most probable letter sequence of a corresponding target word. The each resulted letter of the sequence in each trial group (repetition) is then compared with the actual target letters in a run to calculate the numbers of error made and obtain the performance values of the BCI system.

The procedures of the Viterbi algorithm for a N-th order HMM are provided as a pseudo-code in Algorithm 1.

3.3.3 N-gram Probabilities and Katz Back-off Smoothing

According to the Markov assumption, the future behaviour of a dynamical system only depends on its recent history. In particular, in a kth-order Markov model, the next state only depends on the k most recent states, therefore an N-gram model is a $(N-1)$ -order Markov model [66].

N-gram language modelling is used to estimate the conditional probabilities of a sequence of letters from a corpus based on the relative frequency of letter sequences. Suppose that $l_1 l_2 l_3 ....l_n$ is a letter sequence, N-gram probabilities of this sequence can be computed as follows [67]:

$$P(l_n|l_{n-N+1}, ..., l_{n-1}) = \frac{count(l_{n-N+1}, ..., l_{n-1}, l_n)}{count(l_{n-N+1}, ..., l_{n-1})} \tag{3.36}$$

Algorithm 1 :N-order Viterbi algorithm

procedure VITERBI($\{i_1, i_2, ..., i_n\} \in S$, N is the HMM order)

Initilization:

$\delta_1(i_1) = \pi_i P(x_1|y_1 = i_1)$

$\psi_1(i_1) = 0$

for $n \leftarrow 2, N$ **do**

$\delta_n(i_1, ..., i_n) = \delta_{n-1}(i_1, ..., i_{n-1}) a_{i_1,...,i_n} P(x_n|y_n = i_n)$

$\psi_n(i_1, ..., i_n) = 0$

end for

Recursion:

for $n \leftarrow N+1, T$ **do**

$\delta_n(i_{n-N+1}, ..., i_n) = \max_{i_{n-N}}[\delta_{n-1}(i_{n-N}, ..., i_{n-1}) a_{i_{n-N},...,i_{n-1},i_n}] P(x_n|y_n = i_n)$

$\psi_n(i_{n-N+1}, ..., i_n) = \arg\max_{i_{n-N}}[\delta_{n-1}(i_{n-N}, ..., i_{n-1}) a_{i_{n-N},...,i_{n-1},i_n}]$

end for

Termination:

$c_T = \arg\max_{i_n}[\delta_n(i_{n-N+1}, ..., i_n)]$

$c_{T-1} = \arg\max_{i_{n-1}}[\delta_n(i_{n-N+1}, ..., i_n)]$

$\vdots$

$c_{T-N+1} = \arg\max_{i_{n-N+1}}[\delta_n(i_{n-N+1}, ..., i_n)]$

Backtracking:

for $n \leftarrow T-N, 1$ **do**

$c_n = \psi_{n+N}(c_{n+1}, c_{n+2}, ..., c_{n+N})$

end for

end procedure

The count values in Eq.(3.36) are obtained from a corpus. The training corpus used in this study was first case normalized. Then, only the letter characters was remained in the corpus. A very small number is assigned as the conditional probability for number characters in the speller matrix, which is $\frac{1}{V}$, where V is the total number of characters in the corpus. Using the training corpus, we also learn the conditional probabilities for underline character, "_ ", which denotes the space. The Katz Back-off smoothing technique was used to re-evaluate zero and low-probability n-grams and assign them non-zero values [67]. In this smoothing, lower order model is used when the searched n-gram is unavailable in the corpus for higher order model and the technique recursively backs-off to weaker models until a pre-determined number of count is reached.

$$P_{katz}(w_i|w_{i-N+1},...,w_{i-1}) = \begin{cases} P^*(w_i|w_{i-N+1},...,w_{i-1}), & \text{if } C(w_{i-N+1},...,w_i) > k \\ \alpha_{w_{i-N+1},..,w_{i-1}} P_{katz}(w_i|w_{i-N+2},...,w_{i-1}), & \text{otherwise} \end{cases} \tag{3.37}$$

where N is the order of language model, $C(x)$ is the number of occurrences of letter sequence x in the training corpus, and w_i is the ith letter in a sequence of letters or word. Here, P^* is a discounted probability estimate obtained by Good-Turing estimation [67], k is set to 0, and α is the back-off weight calculated as follows:

$$\alpha_{w_{i-N+1},...,w_{i-1}} = \frac{1 - \sum_{\{w_i:C(w_{i-N+1},...,w_i)>k\}} P^*(w_i|w_{i-N+1},...,w_{i-1})}{\sum_{\{w_i:C(w_{i-N+1},...,w_i)\leq k\}} P_{katz}(w_i|w_{i-N+2},...,w_{i-1})} \tag{3.38}$$

3.4 Channel Selection

In this study, EEG data were recorded by using 10 active electrodes (channels). In order to test the robustness of the proposed language model, channels are eliminated one by one to find M channels performing the best performance out of N, where N is 10 and M is the desired number of channels. The Sequential Floating Forward Selection (SFFS) algorithm [68] is applied as a search algorithm to determine the best subset of channels. The objective function to be maximized is the resulted accuracy value of Viterbi algorithm in the first 5 repetitions obtained from the training data using the features of M number

of channels. Pseudo-code for sequential floating forward selection (SFFS) is provided in Algorithm 2 [69].

Algorithm 2 :Sequential Floating Forward Selection

procedure SFFS(Y_m: channels in the subset, $J(x)$:objective function, $0 \leq m \leq 10$)
    Initiliaze channel set (Step 1):
    $Y_0 = \{\varnothing\}$; $m = 0$
    Select the best channel and update Y_m (Step 2):
    $x^+ = \arg\max_{x \notin Y_m}[J(Y_m + x)]$
    $Y_{m+1} = Y_m + x^+$; $m = m + 1$
    Select the worst channel (Step 3):
    $x^- = \arg\max_{x \in Y_m}[J(Y_m - x)]$
    if $J(Y_m - x^-) > J(Y_m)$ **then**
        $Y_{m+1} = Y_m - x^-$; $m = m + 1$
        Go to Step 3
    else
        Go to Step 2
    end if
end procedure

3.5 Comparison with Relevant Works

Our work is significantly different from previous work in [15,16]. The approach in [15] 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 [15] and [16] 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 [26]. The approach proposed in [15] has been

implemented in this study and the results are compared with the results of our approach and will be presented in Chapter 4.

Martens et al. [24] proposes two decoding systems one of which considers the dependency between the brain response to a target event and brain response to the subsequent stimulus event and the other does not take into account this dependency and overlapping effects. They also incorporate the letter frequency information (prior) to the decoding model and obtain an increase in the decoding performance up to 5% in accuracy since they only focus on modelling the behaviour between the subsequent stimulus events and the corresponding brain signals recorded while that stimulus events occurs. Kindermans et al. [25] perform unsupervised learning on P300 speller incorporating prior information from both inter-subject transfer and n-gram language modelling. They show that the performance of the P300 is improved when the order of the language model (from uni to tri) is increased, as we have also witnessed in this study. They compute the probability of characters given the EEG data by using a forward-backward recursion and plug it to EM algorithm. Although their main goal is to develop an adaptive BCI system without the need of any calibration time, their language model approach utilizing both unsupervised and supervised classifiers overlaps with our work. However, it is necessary to specify that both of these studies were carried out concurrently and independently.

Chapter 4

Offline Analysis

This chapter contains the details of the signal processing procedures applied on the EEG data and presents the experimental results for offline analysis and comparisons with other relevant studies.

4.1 Background

In BCI studies, offline analysis of signals refers to the scenario in which analysis of the recorded signals is performed after the data collection experiment. The situation might even be that experiments are conducted in one laboratory in a broad period of time, and then the analysis is done in another place at a later time with all the experimental data for each subject at once. The reason of this analysis is to develop classification algorithms and techniques as well as to assess and cross-validate known techniques for the aim of obtaining satisfactory performance values for P300-based BCIs [28]. During the recording of the data used for offline analysis, there is no feedback capability to indicate the classification result and show the subject his/her choice. Instead, offline analysis could serve as a first step towards the development of a real-time P300 speller system, where the classification result is displayed to the subject in real-time.

For evaluating and optimizing the performance of our analysis method later to be used in online analysis, we conducted several offline analyses with our own recordings collected in our own laboratory, as well as with a widely used competition data, namely BCI competition II, dataset IIb. The methods and results of these datasets will be presented in the following sections.

4.2 Terminology

To understand clearly the work presented in this thesis, some extensively used terms should be clarified [28].

- A **target letter** is the letter that the subject is informed to focus on at a time instant.
- A **trial** denotes the intensification of each row or column, the timing of which is marked by trigger signals in the recording. We also use the term "flash" in this thesis to imply a trial.
- A **trial group** is the group of trials that include each row and column intensification that is flashed exactly once. For example, with a speller matrix dimension of 6×6, a trial group consists of 12 individual flashes in which there are no rows or columns that are flashed more than once. With this in mind, a trial group is the smallest data set for a P300 classification problem. In this thesis, a trial group is sometimes referred as repetition or one set of flashes.
- An **epoch** is a determined period of recorded data that includes a trial. In P300 studies, this period is usually from 600 ms to 1000 ms starting from the time when a stimulus event (flash) occurs.
- A **run** is the collection of several trial groups. A run is recorded for each letter defined in a session to be spelled. There can be a period of a few seconds between each run, but the recording is not interfered with, and continues.
- A **session** is the time period in which the recordings for all previously defined number of target letters are done.
- A **session group** is the collection of all sessions recorded with one subject during the course of a day with a time break in between each session.

To increase the performance of the classifier, the number of recorded trial groups (repetitions) in a run in the training set has to be increased. However, that leads to low

information rate and therefore the speed of the BCI system is decreased. Hence, researchers on BCI try to come up with efficient signal processing and classification algorithms to achieve performance improvements in accuracy and speed simultaneously.

4.3 P300 Classification Problem

The classification problem in the P300 context is to determine whether the epoch in question contains a P300 wave or not, or in other words, if the stimulus in question creates a P300 wave or not. The first session is recorded for training purposes and thus these data are used as a training set to fit a model that can be used for detection of the test data P300 responses [28].

In the first step of our classification algorithm, the trials are investigated in the recorded data in groups of 12, which is the total number of rows and columns. A classification score is generated according to the distance of data in each epoch to P300 responses in the training set. These scores are separated for columns and rows and trials with maximum scores in each group are selected as the result for the classification process. Since this result includes a row and a column number, the intersection of these row and column gives a single letter in the matrix which is actually the result of the classification algorithm. If this letter is the same with the letter given to the subject as the target, this means the classifier has performed correctly, and we have a correct result.

In the first step of the classification algorithm, we have used Bayesian Linear Discriminant Analysis classifier, which was described in Section 3.1. The algorithm was proposed in [47], and the actual code was developed by Ulrich Hoffmann of the EPFL BCI group in 2006. The classifier uses the first of two sessions as the training set, and the other session is given to the classifier as the test set. The Logistic Regression classifier is also implemented for the first step with its discriminative training on the training set. MATLAB was used to perform all the offline analysis of the experimental data and to read the BDF (BioSemi Data File) file that converts the recorded data into MATLAB format. The software platform also creates the necessary data structures and applies the training and testing of the chosen classifier with the given data. As the data structures are formed after the recorded data file is read, data epochs of 1-second (1000 ms) periods that follow each

trigger signal are extracted and each epoch is labelled as 1 or -1 according to the target stimuli [28].

4.4 Methods

4.4.1 Data pre-processing

The system used the most popular stimulus type, a 6×6 matrix of characters as shown in Figure 1.1. 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 inter-stimulus interval (ISI) of 125 ms: a flash duration of 50 ms and for the remaining 75 ms, all the elements in the matrix dim and the system waits for the next flash. As previously stated, using several number of trial groups in each run increases the performance of the classifier and results in higher accuracy. For this reason, we fixed the maximum number of trial groups for each run to be 15. Due to the size of the matrix, each trial group involves 12 trials, 2 of which are relevant stimuli and 10 of which are irrelevant stimuli. The data pre-processing steps we describe below mainly follow the procedure in [28].

The data are recorded by a BioSemi ActiveTwo system. The data are sampled at 2 kHz (2048 Hz). We use 12 active electrodes in the recordings which are placed in Fp1, Fp2, Fz, Cz, Pz, Oz, P3, P4, PO7 and PO8 locations according to the international 10-20 system, as well as two auxiliary electrodes for reference that are located on the mastoid channels. These mastoid channels are used for reference.

ActiView software saves the recordings as a BDF file , which uses a format originally developed by BioSemi. The recordings saved in a BDF file are turned into a MATLAB file via the code developed by Alois Schloegl in 1998. After this process, the raw data go through several signal processing steps before the classification step.

First, the trigger channel is extracted from the raw data. Times (sample numbers in the sequence) and values (actual trigger values) of each trigger signal are obtained from the data in the trigger channel and stored in a key-value pair. This information will later be used for dividing the data into epochs.

To get better recognition for P300 waves, the data have to be filtered. The whole data

are filtered with a 3^{rd} order Butterworth bandpass filter with cut-off frequencies 1-12 Hz for the aim of reducing the size of the feature space and getting rid of irrelevant frequency components involving background noise and DC offset that occur between electrodes and the skin due to sweating. This filtering removes most of the unwanted artifacts. To obtain a greater SNR, the data have to be re-referenced to a channel or a combination of channels. The data obtained from the mastoid channels are assumed to contain body potentials due to muscle movement and no EEG signals relevant to P300 waves. Hence, by taking the mean of two mastoid channels, a reference signal is acquired and by subtracting these from the other channels, the data are referenced.

Then, we extract the ASCII coded letter trigger signals in the trigger channel. The data cell that holds the runs in a session is sized according to the number of extracted ASCII codes which corresponds to the number of letters in the recording.

In the next step, each run is separated according to the time of appearance of the aforementioned triggers and a standard data structure is formed for each run. A standard data structure for a run involves several parameters, such as a 'target' parameter that keeps the target letter, a 'targetposition' array that represents the position of the target as an array whose elements correspond to a row and column value, a 'stimuli' array that holds the trigger values extracted from the trigger channel, a 'labels' array for deciding if the elements in 'stimuli' array involves the target letter or not, a 'times' array that holds the sample times (sample numbers in the sequence) indicating when an element in the 'stimuli' array has occurred. The principal trigger signals are positions of intensifications. Each row or column has a unique ID number, and whenever a row or a column is highlighted, the corresponding ID is sent over the trigger channel to the recording device. Rows and columns are numbered from 0 to 11, where columns are numbered from 0 to 5 and rows are numbered from 6 to 11. These trigger values are stored in the 'stimuli' array as mentioned above.

The next preparation step is forming the epochs from the whole data by utilizing the values in the 'times' array. Each epoch is a data set of 1 second (1000 ms). Since the data are digitized at 2 kHz, each epoch holds 2048 samples. To reduce the size of the feature space and remove the unnecessary features, the data are decimated by 64.

In the next step, the data are normalized to remove the negative effects of electrode-skin resistance that result in amplitude changes and other anomalies. However, if the waveform involves very high and extreme values, normalization may result in a poor performance. To avoid this problem, the data are windsorized in a 10% window. Windsorizing the data removes the extremities by clipping the samples that are out of this window and provides a healthy normalization [28].

At last, the data are ready for classification. In the final state, epochs are represented as an $m \times n \times t$ matrix where m is the number of samples per epoch after decimation, which is 32, n is the number of channels (10 in this case) and t is the number of epochs (180 for 15 trial groups).

4.4.2 Classification

After the data pre-processing procedure, we obtain the feature vector for each epoch by concatenating the filtered data from each electrode, i.e., a vector of 320 samples for the case of 10 electrodes. At the end, the data are reshaped as a matrix of size $r \times t$ where r is the size of the feature vector (320 samples), and t is the number of feature vectors (epochs). This data sample are run through a classification process described below.

As it was mentioned in Section 4.3, BLDA calculates a score for each element (epoch) in the test set reflecting its similarity to the attended class. After that, the scores for epochs grouped in sizes of a trial group will be calculated. Since the matrix dimension is 6×6, the epochs are handled in groups of 12. For example, if we consider a complete run as a single test set, there are 180 epochs in it. A total of 15 trial groups are generated and rows and columns with the highest score are obtained as P300 detections. For a purely-data driven approach, if the position of the target corresponds with these row and column values, the classifier result is counted as a correct. In the language-model based approach we propose, the scores for each letter obtained through this process rather than the classification outputs are used.

The scores obtained for the letter at times instant i as described here are used as "data" x_i by the language model-based classification approach described in Section 3.3. The final language model-based classification output is obtained by the Forward, Forward-

Table 4.1: Target words of our own Dataset.

Session	Target words
Training	KALEM_YOLCULUK
Test	KITAP_MASA_AGLAMAK_SIKINTI

Backward, or Viterbi algorithms as described in Chapter 3.

4.5 Experiments

4.5.1 Datasets

The first dataset used in this study includes the offline spelling data recorded in our own laboratory by 7 healthy subjects, whose ages varied between 18 and 30. Only two of the subjects had previous BCI experience. The used electrode sets, stimulus timing (ISI) and maximum number of repetitions for this data were already described in Section 4.4.1. Each subject participated two sessions: a training session and a 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. Table 4.1 shows all the target letters chosen to be spelled by the subjects.

The second dataset consists one of most widely used datasets in BCI research: BCI Competition II Dataset IIb [70]. Our goal is to assess the performance of our method on this dataset and compare it with the methods that obtained the best results on this data in the competition. The data are collected from one subject in three sessions, and sampled at 240 Hz and recorded using 64 channels. Each session consisted of a number of run sets. In each run set, the subject focused attention on a series of characters (target word). The training data consists of 11 target words totally consisting of 42 letters. The test set features 7 target words composed of 27 letters. Only one word is used two times in the sessions. For each character, user display was as follows: the matrix was displayed for a 2.5

Table 4.2: Target words of BCI Competition II Dataset IIb.

Session	Target words
Training 1	CAT DOG FISH WATER BOWL
Training 2	HAT HAT GLOVE SHOES FISH RAT
Test	FOOD MOOT HAM PIE CAKE TUNA ZYGOT

s period, and during this time each character had the same intensity (i.e., the matrix was blank). Subsequently, each row and column in the matrix was randomly intensified for 100 ms. After intensification of a row/column, the matrix was blank for 75 ms. Row/column intensifications were block randomized in blocks of 12. Sets of 12 flashes were repeated 15 times for each character. Each sequence of 15 sets of flashes was followed by a 2.5 s period, and during this time the matrix was blank. In this period, the user was informed about the next character to be typed in the word. Table 4.2 shows all the target letters for each session used in this dataset.

4.5.2 Experimental Results

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 character classifications in a session by the total number of classifications. In order to assess the speed of the communication , information transfer rate, bit-rate (B), in bits/min, is also computed as in [71]:

$$B = \frac{60}{T}\left(\log_2(n) + p\log_2(p) + (1-p)\log_2\left(\frac{1-p}{n-1}\right)\right) \tag{4.1}$$

where p is the accuracy of the classification , n is the number of characters in the speller matrix (36 in this case) and T is the time in seconds that is needed to spell one symbol calculated by $(3.5 + 0.125 \times 12 \times N_t)$, where N_t is the number of available trial groups. Since one set of flashes takes 1.5 s and assuming that 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, which is calculated by $12 \times \log_2 36$, for the first dataset. In a similar way, for the BCI competition dataset, one set of flashes takes 2.1 s. If we take into account the 2.5 s period for the target letter displaying, at most 13.04 characters can be typed in a minute by the subject. Hence, the maximum bit-rate for offline classification of the second dataset is 67.42 bits/min.

Six different methods for classification analysis are compared in this thesis: a general BLDA and LR method that does not use any type of language modelling for letter classification; the "NLP" method, proposed in [15], that develops a language model which only depends on the integration of the EEG scores at the current time with letter probabilities based on decisions of the previous time; and Forward, Forward-backward, and Viterbi methods that utilize a language model incorporated into the proposed HMM as proposed in this thesis, which were already described in Section 3.3. Note that the proposed approach in [15] was originally called the "NLP" method in that paper, hence we will use the same name throughout this thesis.

Classification performance of proposed language model based algorithm using BLDA and LR

We perform training on each subject's EEG data using BLDA and LR classifiers, respectively. As it was mentioned before, in BLDA, the resulted scores are converted into probability values with the use of Gaussian estimation whose parameters are estimated by the training data. On the other hand, LR performs a discriminative training by using training data to directly estimate posterior probabilities. Table 4.3 and 4.5 show the result based on each method. In Table 4.3, all classification algorithms whether utilising a language model or not, use BLDA classifier in the classification of P300 signals, while in Table 4.5, all the algorithms use the LR classifier. The order of the language model is set to trigram for the ease of comparison.

Since speed is very important for the effectiveness of real-time BCI communication, we present performance values for each subject using the first 3 trial groups (repetitions) only in the tables, rather than considering up to 15 trial groups. This provides a comparison

Table 4.3: Performance values for each subject obtained in the first 3 trial groups using BLDA in all approaches.

	Classification Accuracy (%)					**Bit-rate (bits/min)**				
Subjects	BLDA	NLP	F	F-B	Viterbi	BLDA	NLP	F	F-B	Viterbi
S1	50	57.69	61.53	73.08	73.08	12.04	15.12	16.76	22.11	22.11
S2	53.84	80.77	84.62	88.46	84.62	13.55	23.75	28.21	30.47	28.21
S3	57.69	73.08	73.08	73.08	76.92	15.12	22.12	22.12	22.12	24.05
S4	30.77	30.77	34.62	46.15	34.62	5.46	5.46	6.64	10.59	6.64
S5	84.61	88.46	88.46	92.31	92.31	28.21	30.47	30.47	32.88	32.88
S6	69.23	76.92	88.46	88.46	84.62	20.26	24.05	30.47	30.47	28.21
S7	65.38	80.77	88.46	92.31	88.46	18.47	23.75	30.47	32.88	30.47
Avg	58.79	69.78	74.18	**79.13**	76.38	16.15	20.78	23.59	**25.93**	24.77

of methods in the high-speed regime. Table 4.3 shows accuracy and information transfer rate results of all algorithms for each subject and on average. If we compare the proposed methods' results with the BLDA method, we observe the significant improvements both in accuracy and in bit-rate. To be more precise, the overall improvement from BLDA to the F-B algorithm is 34.6% (p= 0.0004, see Table 4.4) and 60.6% (p= 0.0007, see Table 4.4) for accuracy and bit-rate, respectively. The improvements are 29.9% (p= 0.001) and 53.4% (p= 0.001) for the Viterbi algorithm. If we ignore the time for displaying the target letter to the subject, new bit-rate values become (27.7, 44.8, 42.27) bits/min for BLDA, F-B and Viterbi methods, respectively. Table 4.4 shows that all the methods using language model produce statistically significant results.

Table 4.4: The resulted p-values when using BLDA classifier in all approaches.

Method pairs	**Accuracy**	**Bit-rate**
BLDA & F	0.003	0.004
BLDA & F-B	0.0004	0.0007
BLDA & Viterbi	0.001	0.001
NLP & F	0.015	0.019
NLP & F-B	0.002	0.002
NLP & Viterbi	0.003	0.0018

Results in Table 4.5 suggest that our approach provides even higher performance improvements when we utilize discriminative training through LR on the EEG data. We also note that use of LR rather than BLDA leads to slightly worse results in all methods considered. Furthermore, our model outperforms the NLP method of [15] both in accuracy and speed. Table 4.5 demonstrates that in a discriminatively trained generative model, the overall improvements between NLP and F-B methods are 14.9% (p= 0.014, see Table 4.6) and 25.8 % (p=0.006, see Table 4.6) for accuracy and bit-rate, respectively. This 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.

For the sake of space and repetition, we only present the results with using one classifier in all approaches for each subject. The figures from 4.1 to 4.11 represent the performances for each subject either with BLDA or LR classifier as well as the average performances over 7 subjects considering more trial group numbers (max 15). Figure 4.8 asserts that on average, the proposed model utilizing fourgram language model needs at least 5 stimulus repetitions to reach a 90% accuracy while BLDA method can achieve this after 11 stimulus repetitions and NLP need 7 trial groups for this. Since one trial group lasts 1.5 s when we ignore the time for displaying target, our approach will estimate the target letter with

Table 4.5: Performance values for each subject obtained in the first 3 trial groups using LR in all approaches.

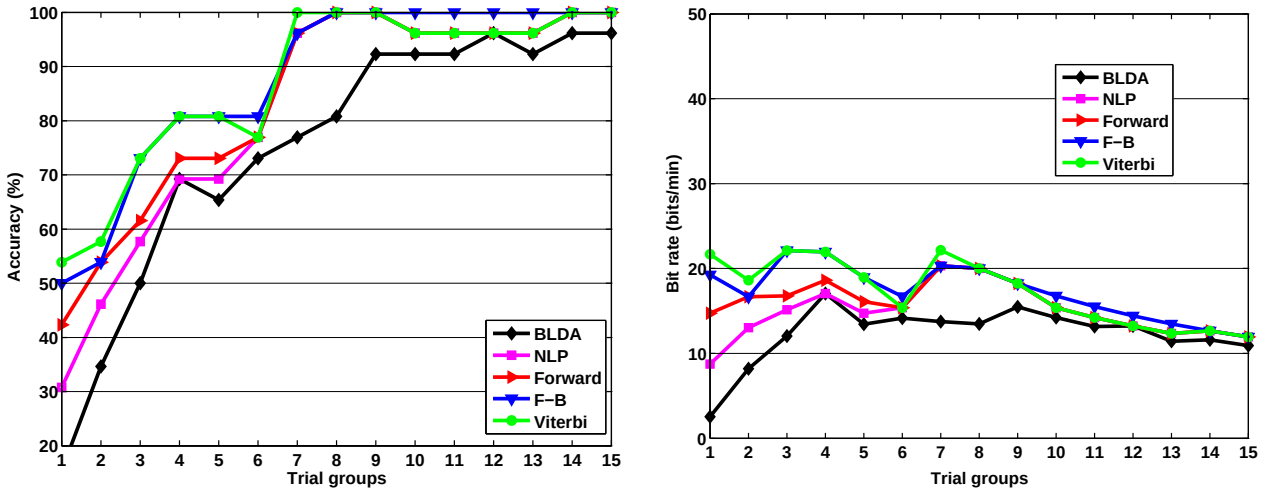
	Classification Accuracy (%)					**Bit-rate (bits/min)**				
Subjects	LR	NLP	F	F-B	Viterbi	LR	NLP	F	F-B	Viterbi
S1	50	46.15	69.23	73.08	69.23	12.04	10.59	20.26	22.11	20.26
S2	53.84	76.92	73.08	84.62	73.08	13.55	24.05	22.11	28.21	22.11
S3	57.69	69.23	76.92	76.92	69.23	15.12	20.26	24.05	24.05	20.26
S4	26.92	38.46	34.62	34.62	26.92	4.36	7.89	6.64	6.64	4.36
S5	76.92	84.62	92.31	92.31	92.31	24.84	28.21	32.88	32.88	32.88
S6	61.54	73.08	76.92	84.62	80.77	16.77	22.11	24.05	28.21	26.08
S7	61.54	76.92	84.62	88.46	80.77	16.77	24.05	28.21	30.47	26.08
Avg	55.49	66.48	72.53	**76.38**	70.33	14.78	19.59	22.62	**24.65**	21.72

90 % classification accuracy 9 s earlier than BLDA does. The corresponding maximum numbers of letters that can be typed with 90% accuracy in a minute are 8 and 3.63 for F-B and BLDA method, respectively if we neglect the time for displaying the answer of the classification. Figure 4.8 (b) illustrates the remarkable effect of our model on the speed of the BCI system particularly in the first three trial groups. The maximum reached bit-rate value in this interval is 16.42 bits/min for BLDA while it is 27.86 bits/min for Viterbi, where the improvement is around 70%.

The results in Figure 4.9 and Figure 4.11 also support the observation we mentioned above that LR classifier performs worse than BLDA method. Although a satisfactory improvement is achieved when small numbers of stimulus repetitions (hence limited data) are available, on average, the improvement closes to zero when higher trial groups numbers are reached as opposed to what we observe in Figure 4.8 (a). Having observed this in average results shown in Figure 4.9, we can not say that the above mentioned situation is valid for all subjects. Figure 4.2 (a) clearly shows that the improvement from LR to language model algorithms are also visible (not close to 0) for higher number of stimulus repetitions, which demonstrates that this case is subject-specific.

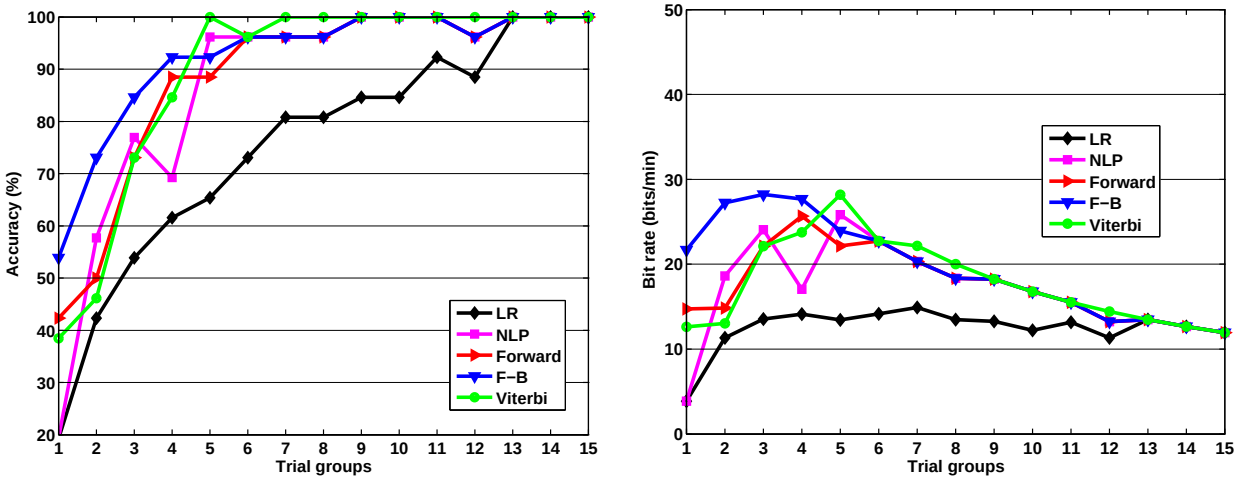
Table 4.6: The resulted p-values when using LR classifier in all approaches.

Method pairs	Accuracy	Bit-rate
LR & F	0.00004	0.00015
LR & F-B	0.00018	0.0003
LR & Viterbi	0.00078	0.0008
NLP & F	0.06	0.044
NLP & F-B	0.014	0.006
NLP & Viterbi	0.19	0.12



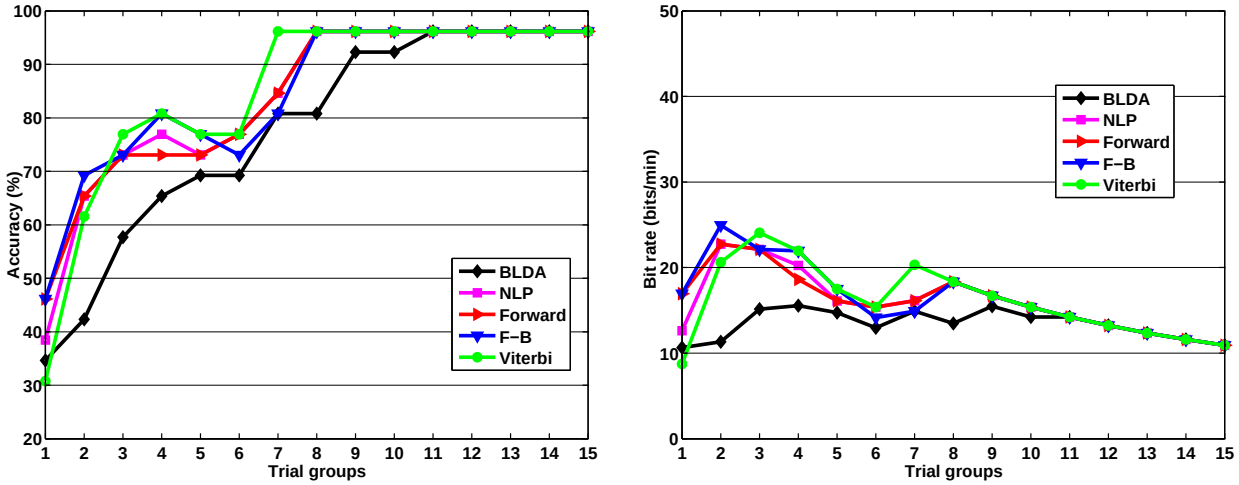
(a) (b)

Figure 4.1: Offline analysis results for subject 1. (a) Accuracy and (b) Bit-rate versus the number of trial groups.



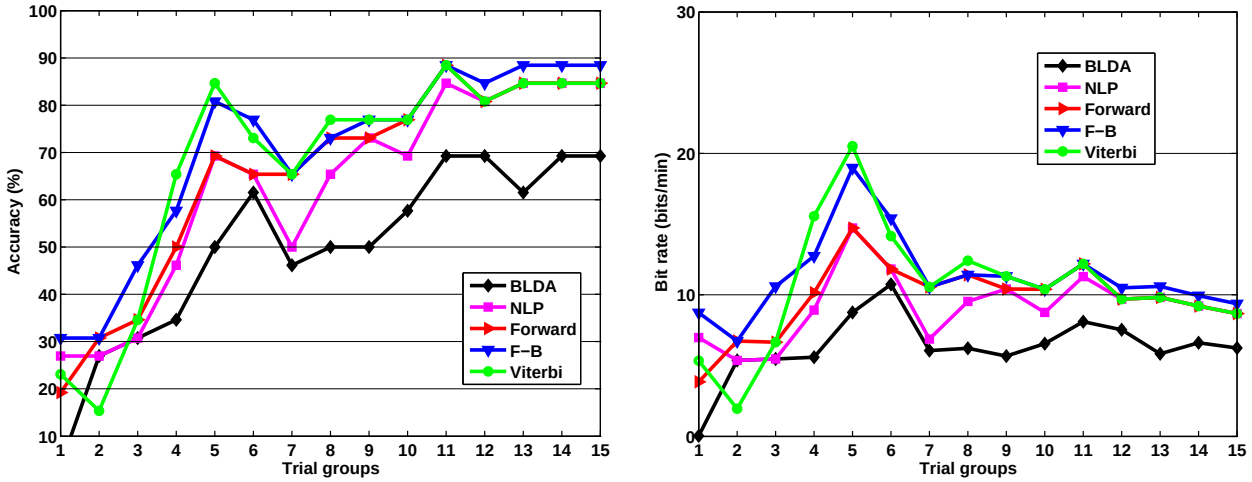
(a) (b)

Figure 4.2: Offline analysis results for subject 2. (a) Accuracy and (b) Bit-rate versus the number of trial groups.



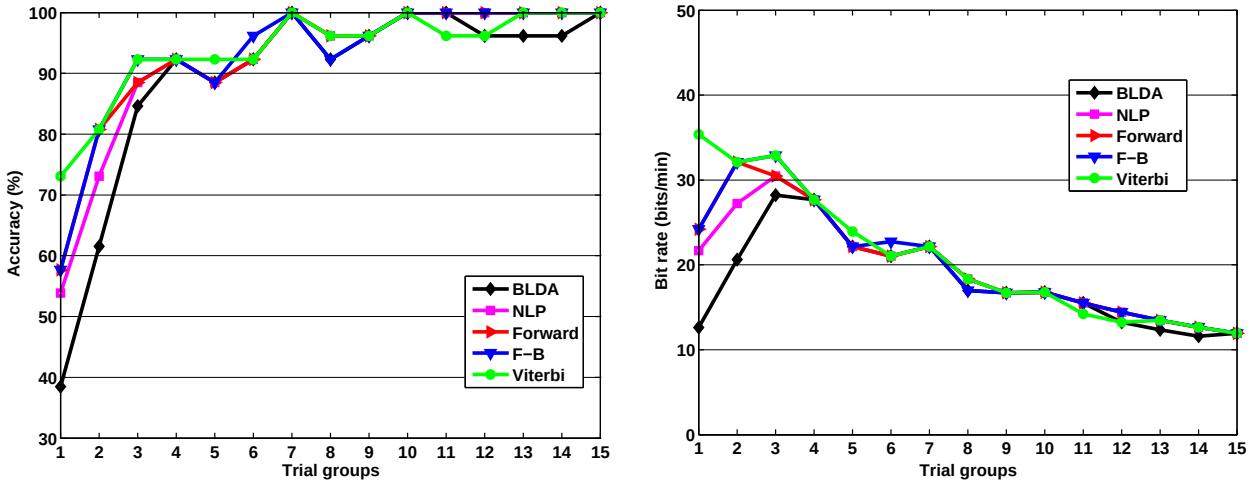
(a) (b)

Figure 4.3: Offline analysis results for subject 3. (a) Accuracy and (b) Bit-rate versus the number of trial groups.



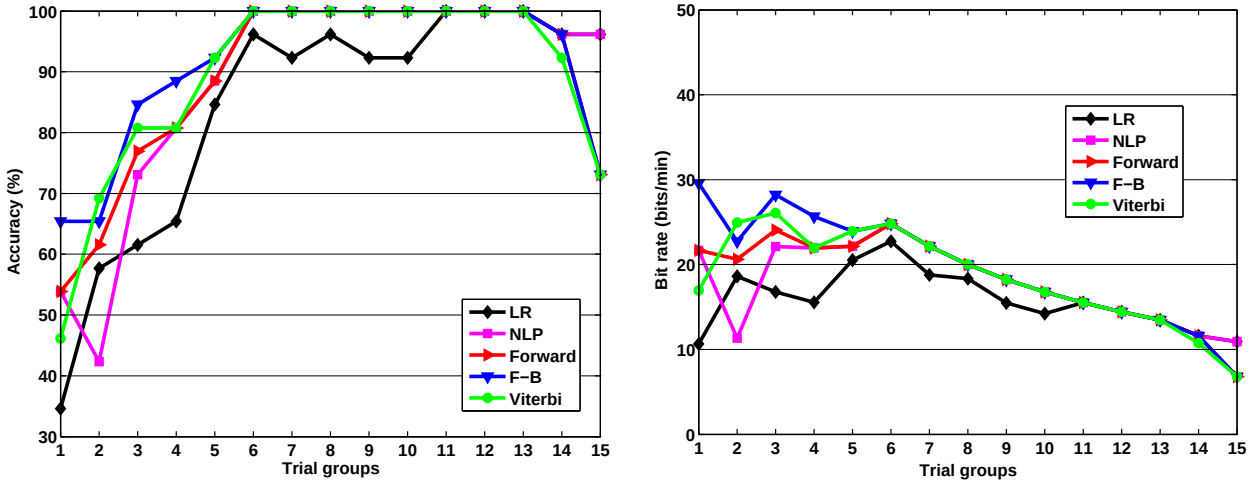
(a) (b)

Figure 4.4: Offline analysis results for subject 4. (a) Accuracy and (b) Bit-rate versus the number of trial groups.



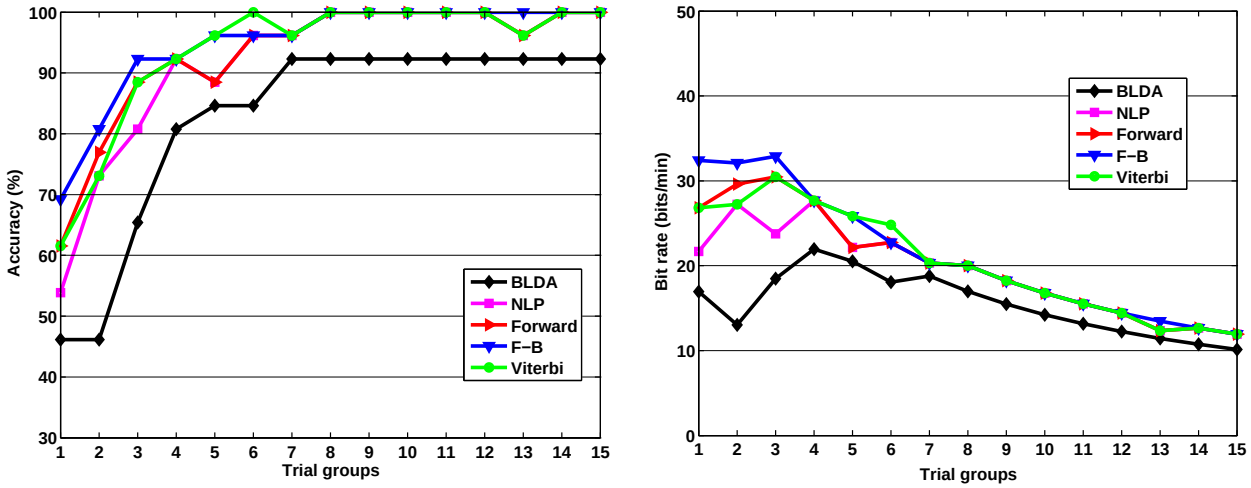
(a) (b)

Figure 4.5: Offline analysis results for subject 5. (a) Accuracy and (b) Bit-rate versus the number of trial groups.



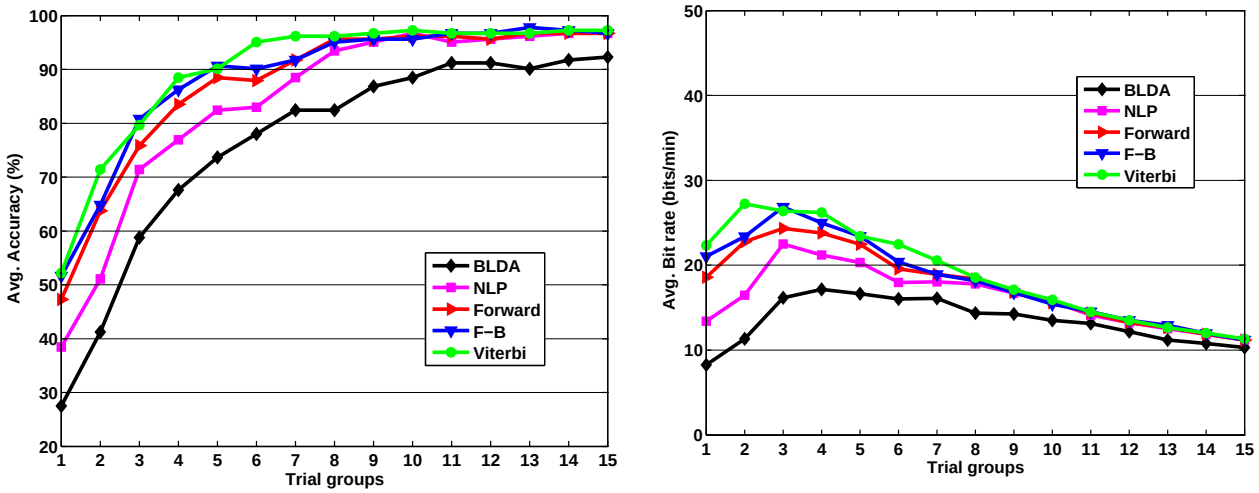
(a) (b)

Figure 4.6: Offline analysis results for subject 6. (a) Accuracy and (b) Bit-rate versus the number of trial groups.



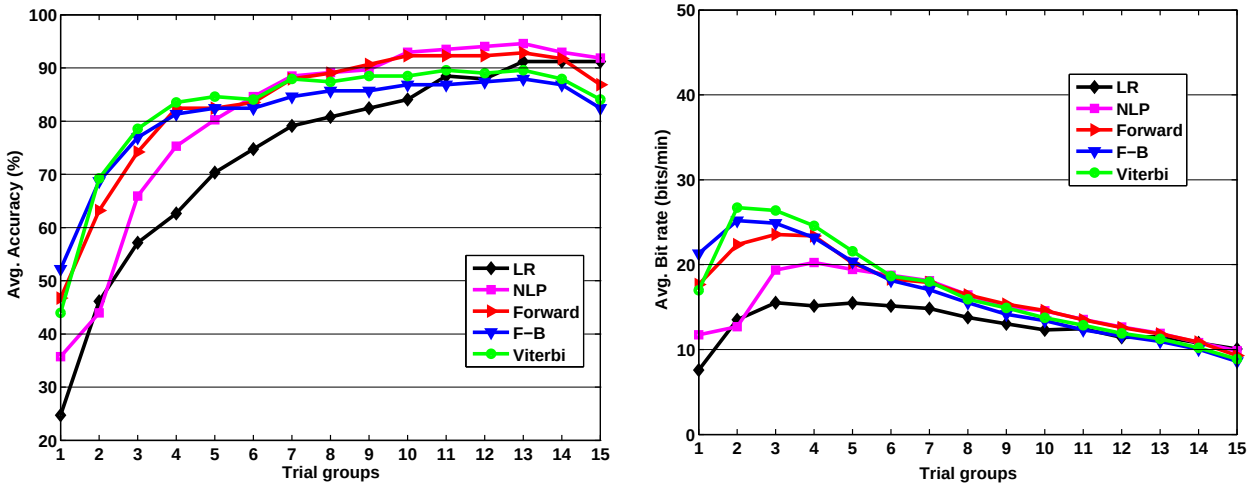
(a) (b)

Figure 4.7: Offline analysis results for subject 7. (a) Accuracy and (b) Bit-rate versus the number of trial groups.



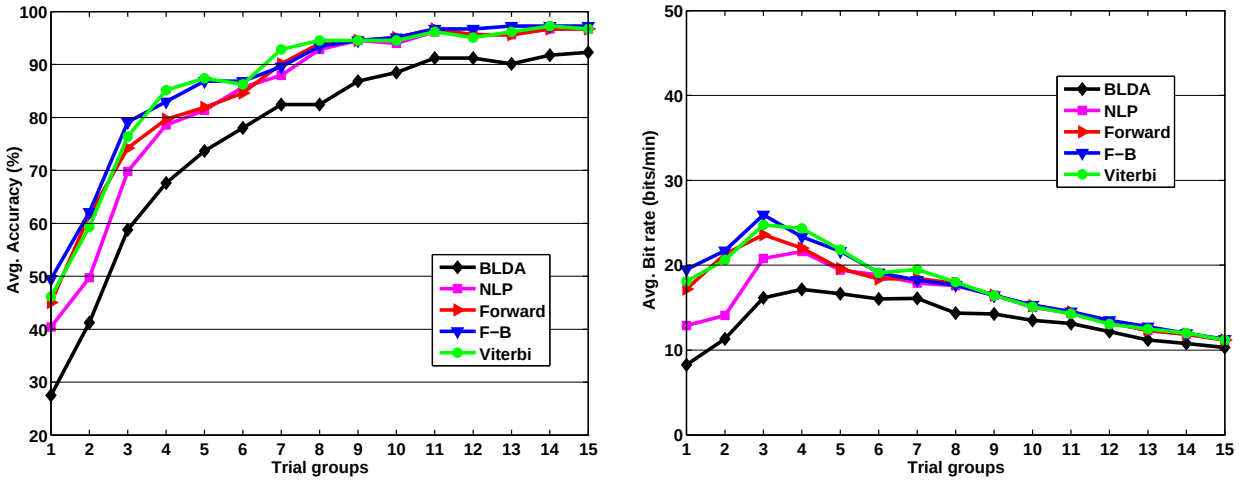
(a) (b)

Figure 4.8: Average classification performance over 7 subjects using fourgram language model with the BLDA classifier. (a) Accuracy and (b) Bit-rate versus the number of trial groups.



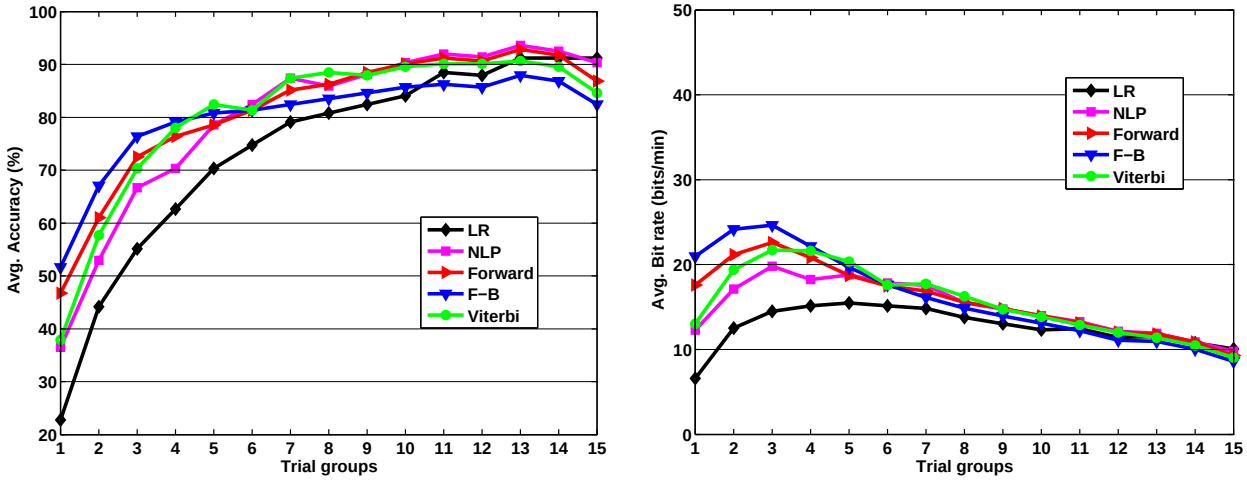
(a) (b)

Figure 4.9: Average classification performance over 7 subjects using fourgram language model with the LR classifier. (a) Accuracy and (b) Bit-rate versus the number of trial groups.



(a) (b)

Figure 4.10: Average classification performance over 7 subjects using trigram language model with the BLDA classifier. (a) Accuracy and (b) Bit-rate versus the number of trial groups.



(a) (b)

Figure 4.11: Average classification performance over 7 subjects using trigram language model with the LR classifier. (a) Accuracy and (b) Bit-rate versus the number of trial groups.

N-grams performance comparison

In this study, we have also considered using different n-gram language models. Unigram, bigram, trigram and fourgram language models are implemented on the proposed model and the performance results are obtained. As shown in Table 4.7, improvements achieved by our proposed methods are more pronounced when the order of the language model is increased. For each n-gram except unigram, the maximum performance values are achieved by either F-B or Viterbi algorithm as shown in bold values in Table 4.7 and 4.8. For the unigram case, the performance values for all language model based approaches are same since unigram language model only utilizes the letter prior probabilities rather than conditional letter likelihoods. Results in Table 4.8 suggest that similar observations can be made when LR is used as the base classifier for the case of 5 trial groups. Both Table 4.7 and 4.8 demonstrate that there is no consistent pattern on the relative performances of F-B and Viterbi methods. This is an expected behaviour. In our work, we observe that F-B or Viterbi methods can outperform each other in different situations. The main reason of this is the difference in the objectives of the two methods. While the F-B algorithm aims to maximize the marginal probabilities of a letter sequence, the Viterbi algorithm maximizes the joint probabilities and therefore tries to find the single best letter sequence.

Table 4.7: Average performance values for different n-grams obtained in the first 5 trial groups using the BLDA classifier

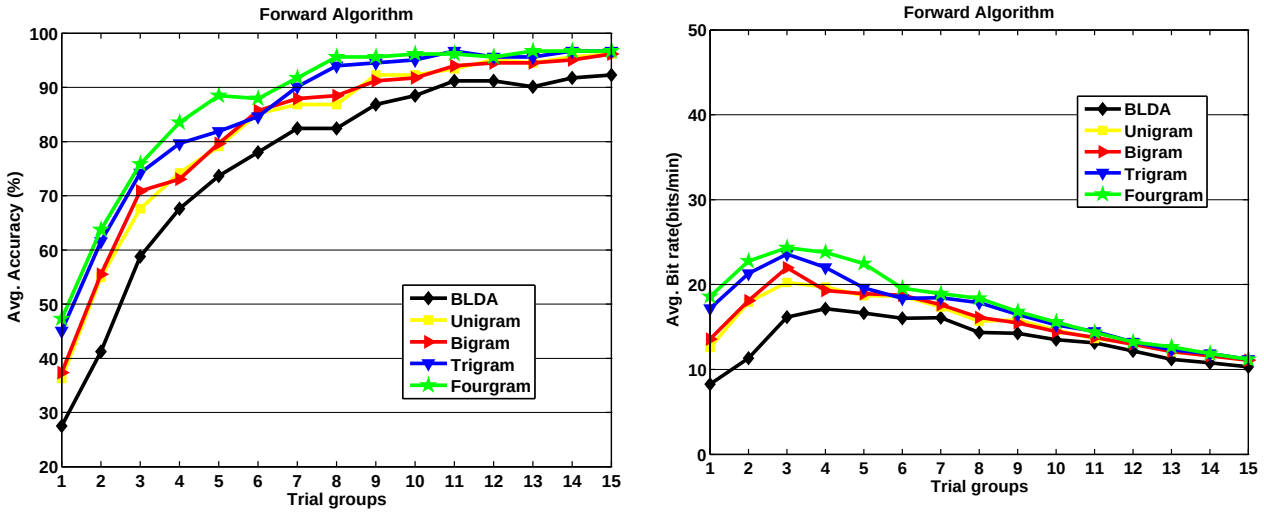
	Classification Accuracy (%)				**Bit-rate (bits/min)**			
Methods	Uni	Bi	Tri	Four	Uni	Bi	Tri	Four
BLDA	73.62	73.62	73.62	73.62	16.64	16.64	16.64	16.64
NLP	79.12	78.57	81.31	82.41	18.67	18.45	19.41	20.31
Forward	79.12	79.67	81.87	88.47	18.67	18.92	19.61	22.47
Forward-Backward	79.12	**81.32**	86.83	**90.66**	18.67	**19.56**	21.63	**23.42**
Viterbi	79.12	80.22	**87.36**	90.11	18.67	19.05	**21.83**	23.40

To illustrate the effect of the order of the language model on the performances, the average accuracy and bit-rate with respect to trial group numbers of F, F-B and Viterbi

methods using either BLDA or LR classifier are provided in Figure 4.12, 4.13 and 4.14. From the figures, one can recognize the effect of the higher order language model on the speed and accuracy of the system especially when small number of data (sets of flashes) are available.

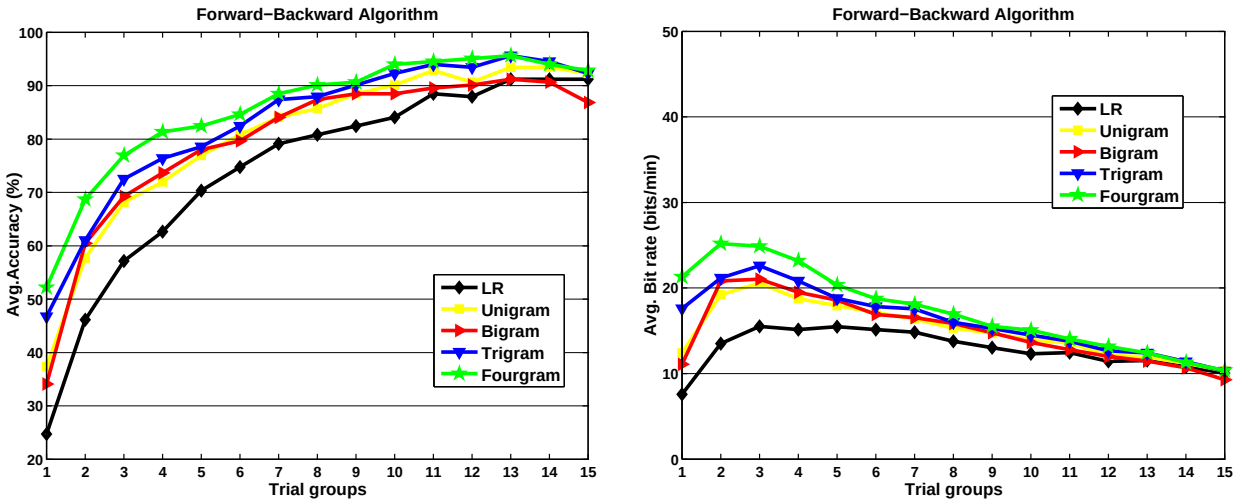
Table 4.8: Average performance values for different n-grams obtained in the first 5 trial groups using the LR classifier

Methods	**Classification Accuracy (%)**				**Bit-rate (bits/min)**			
	Uni	Bi	Tri	Four	Uni	Bi	Tri	Four
LR	70.33	70.33	70.33	70.33	15.50	15.50	15.50	15.50
NLP	76.92	78.02	78.57	80.22	17.92	18.58	18.79	19.43
Forward	76.92	75.28	78.58	82.42	17.92	18.02	18.71	20.17
Forward-Backward	76.92	**79.67**	80.77	82.42	17.92	**19.07**	19.66	20.35
Viterbi	76.92	79.12	**82.42**	**84.62**	17.92	18.94	**20.34**	**21.59**



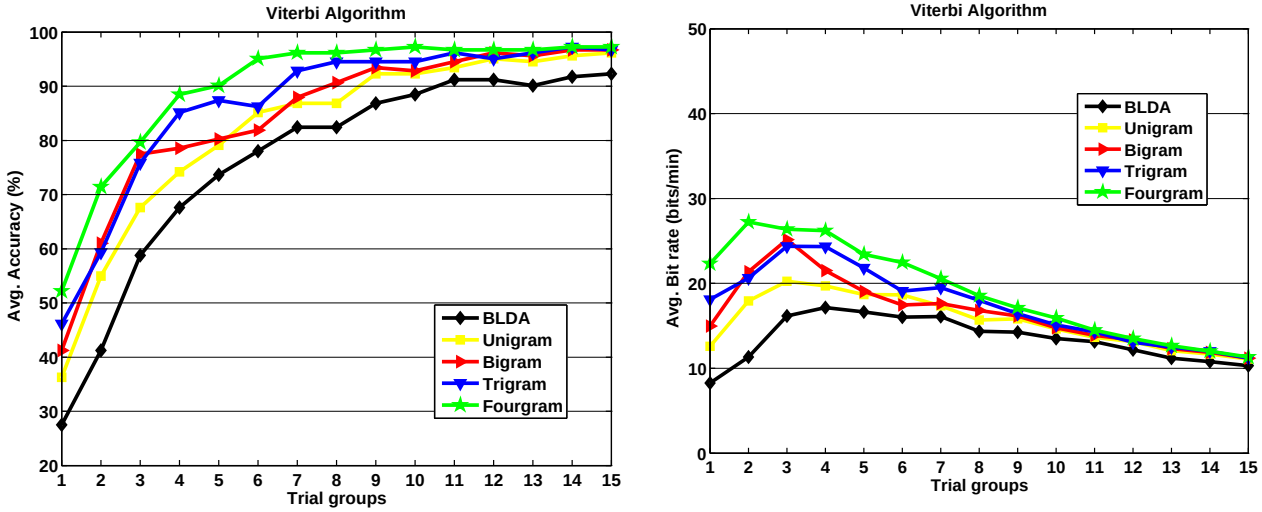
(a) (b)

Figure 4.12: (a) Average accuracy and (b) Average bitrate versus the number of trial groups for different n-grams using the Forward algorithm with the BLDA classifier.



(a) (b)

Figure 4.13: (a) Average accuracy and (b) Average bitrate versus the number of trial groups for different n-grams using the F-B algorithm with the LR classifier.



(a) (b)

Figure 4.14: (a) Average accuracy and (b) Average bitrate versus the number of trial groups for different n-grams using the Viterbi algorithm with the BLDA classifier.

Channel selection results

We now evaluate the robustness of our proposed methods to limitations in the quantity of data. To this end, we consider using smaller than the available 10 channels (electrodes). The training data of the each subject are used to find the best M number of electrodes where $1 \leq M \leq 10$. As a result, the subject specific electrode subsets are obtained from the training data of each subject and all the algorithms are implemented with that electrode subset. Only BLDA classifier is used in channel selection and trigram language model is employed in the classification algorithm.

Figure 4.15 presents performance results averaged over all subjects versus the number of electrodes. Results are promising to show the robustness of the proposed classification algorithms. If the data from a small number of electrodes rather than 10 is used, the relative accuracy improvements for proposed methods are much better as shown in Figure 4.15 (a) and (c). F-B and Viterbi algorithm may reach to 90% accuracy in the first 5 trial groups just only using 5 electrodes while the BLDA method can not manage it in the first 5 repetitions with any number of electrode sets. Furthermore, our model can exhibit better performance compared with the NLP method for any number of available electrodes.

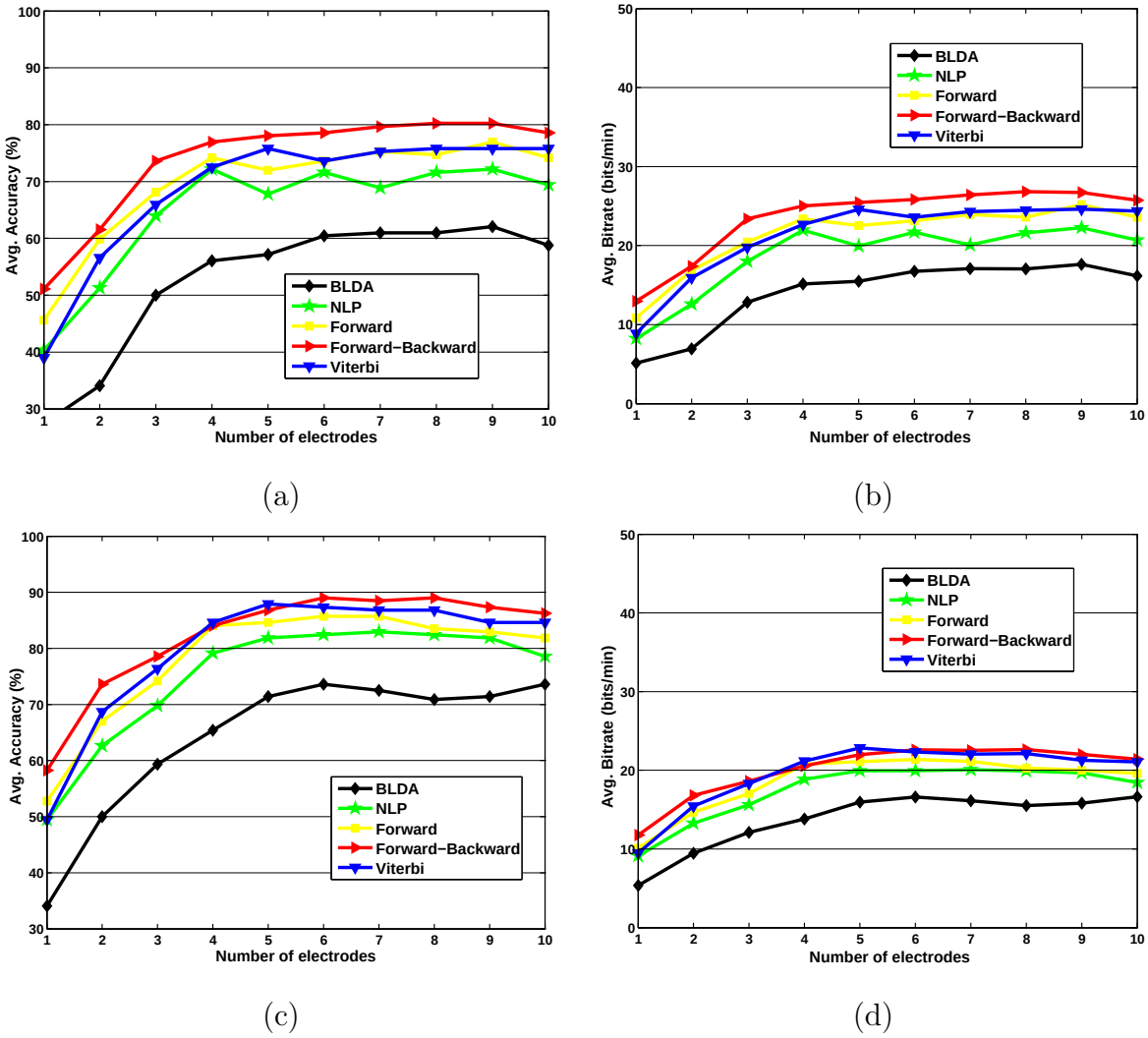


Figure 4.15: (a)-(b) Average accuracy and bit-rate versus the number of electrodes obtained in the first 3 trial groups,(c)-(d) Average accuracy and bit-rate versus the number of electrodes obtained in the first 5 trial groups.

BCI Competition Dataset II Results

Before applying the proposed algorithm to the BCI Competition II Dataset IIb, various data pre-processing techniques were implemented in order to get better recognition of the P300 component based on this dataset whose data acquisition features, time and channel parameters were already described in Section 4.5.1. An English dictionary corpus [72] including approximately 250,000 distinct words were used to obtain the letter conditional probabilities of the language model. The procedures that were followed for the

classification of this dataset are listed below.

- 12 electrodes out of 64 were chosen from the position of central, parietal, and occipital lobes where the P300 signal is known to be more apparent. The selected channels are Fz,Cz,Pz,Oz,O1,O2,P3,P4,CPz,FCz,PO7 and PO8. Note that 8 of these channels were also used in our own EEG recordings.

- A Common Average Reference (CAR) was applied. First, the average of all 64 channels signal is computed and then subtracted from each channel [73]. Unlike our recordings, the mastoid channels were not used in this dataset for reference purpose.

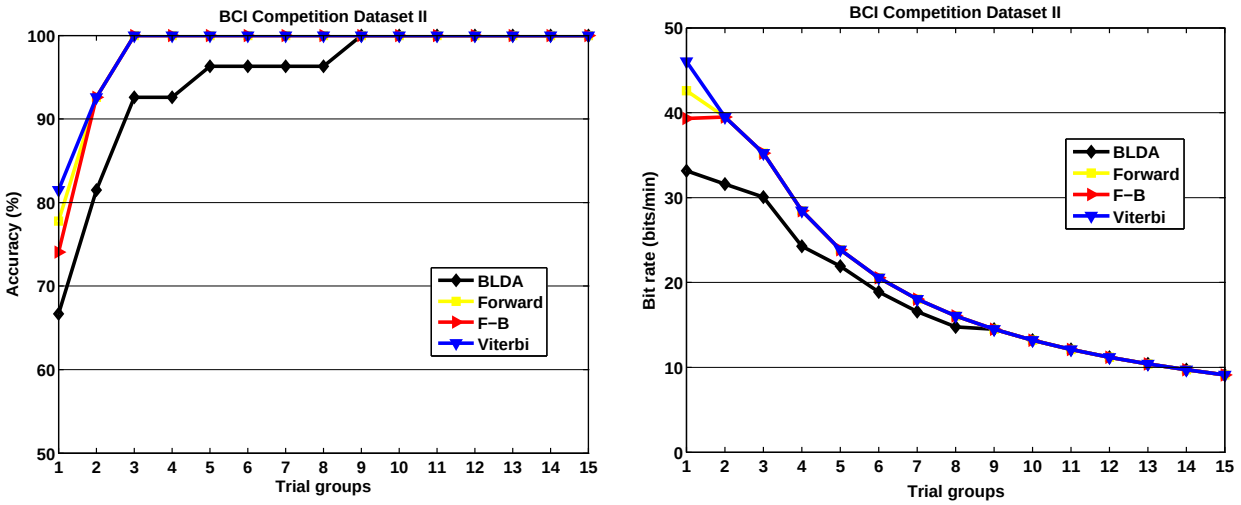
- A band-pass filter with cut off frequencies 1-30 Hz was applied to the whole data and the data were normalized and windsorized.

- A time frame of 0-667 ms post stimulus, starting from the sample where the stimulus is presented, was extracted to constitute an epoch. The dimensionality was reduced by sub-sampling with a factor 8 and 20 samples for each channel was retained.

- The proposed model using BLDA classifier is applied with the trigram language model.

Figure 4.16 shows the obtained performance values after applying the above steps to this dataset. BLDA method achieves 100 % accuracy (perfect classification) using all 9 stimulus repetitions out of the available 15. By using our proposed model, this minimum number of repetitions becomes just 3, which is really a significant improvement in terms of speed for perfect classification. To be more precise, on average our model reaches 100 % accuracy 6 stimulus repetitions before the BLDA method, which corresponds 12.6 s earlier per character (without considering any time used to display the classification results). Hence, the maximum number of letters that can be typed with perfect classification is 9.52 letters per minute with the proposed model while this number is only 3.17 letters per minute for the BLDA method.

The winner of the BCI Competition II Dataset IIb manages to obtain perfect classification at minimum 5 stimulus repetitions. Our result shown in the Figure 4.16 demonstrates that the proposed model outperforms the value achieved by the winner [70]. We should



(a) (b)

Figure 4.16: BCI Competition Dataset II offline analysis results. (a) Accuracy and (b) Bit-rate versus the number of stimulus repetitions.

note that our classification process does not use any information that was unavailable to the competition participants, hence our comparison is fair. Table 4.9 presents obtained accuracy values by using all 15 stimulus repetitions, and the minimum number of repetitions needed for perfect classification by our approach as well as the top performing competition participants. These results suggest that our approach reaches the desired accuracy faster than the approaches that took part in the competition thanks to the information it exploits based on the learned language model.

Table 4.9: Performance values in literature for BCI Competition Dataset II

Contributor	**Acc.(%)**	**Min.repetition**
Matthias Kaper [18]	100	5
Xiaorong Gao [74]	100	5-8
Vladimir Bostanov [75]	100	6
Benjamin Blankertz [76]	100	6-11
David Tax [76]	100	n/a
This work	100	**3**

Chapter 5

Online Spelling Experiments

This chapter focuses on online recording analysis and feedback of P300 signals. In chapter 4, we explored the capabilities of offline analysis, where the brainwaves are recorded first and the analysis is performed separately afterwards, allowing one to do various kinds of analyses on the data. In online analysis, instead, brainwaves are pre-processed and analysed concurrently, as the recording continues. The result of the classification is displayed to the subject for each time instant.

5.1 Background

During the online spelling analysis, the data are filtered and processed in real time, as they arrive. In other words, the data are first divided into epochs and then filtered and processed locally in epochs, instead of filtering and processing as a whole and then dividing into epochs, which is what we did during the offline analysis. As it was discussed in detail in [28], filtering just a part of data separately instead of filtering the data in whole produces a problem in dealing with low frequency components, as they are reflected in the recording as offset drifts. Filtering the data as a whole helps one to have more samples, hence even the smallest frequency component may complete a period whereas in filtering by epochs, one cannot decide whether there is a sub-Hertz component or not. With this in mind, it was clearly shown in [28] that the order of filtering and epoch extraction effects the performance of the classification. Although the order of the filtering (before or after epoch extraction) can be chosen freely for offline analysis, filtering has to be done in-epoch for online analysis. Consequently, the study in [28] demonstrates that most of the time,

online analysis results in worse performance than the offline analysis with actually the same data.

5.2 Method

In this study, several number of online spelling sessions were carried out to observe the improvements that was achieved by online analysis. For this reason, seven healthy subjects participated online spelling. All the subjects participated three sessions : one training and two test sessions. In the training session, the subjects spell previously defined two words involving totally between 10 and 20 characters and in each one of the two test sessions, the subjects aim to spell the same target word groups that are freely selected by them before the beginning of the test sessions. The target words were chosen from the words listed in table 4.1 for each subject and the same words were not necessarily used for all subjects. The maximum number of trial groups for each run of the training session was defined according to subject's request. It can be either 5,10 or 15. In addition to this, we used a different flashing paradigm where the letters flash in a randomly coloured fashion. We believe this stimulation technique will elicit a bigger P300 response in the subject's brain, because the subject might get used to white flashes and expectations might arise. In a randomly coloured fashion, there are two surprising events; one, as usual, the subject is unaware of when the flash will happen, and two, the subject is unaware of in what color the flash will happen [28].

5.2.1 Data pre-processing

As we have mentioned before, the data are processed in-epoch. That is, incoming data are divided into relevant epochs first. After enough data to fill an epoch is streamed in, that epoch is ready for data preparation.

In the first step, the data are bandpass filtered in a 1-12 Hz band-pass to remove unwanted frequency components. Next, they are re-referenced to the mean of two mastoid channels. The data are then windsorized in a 10% window and normalized, and decimated by 64 [28].

The data pre-processing, stimulus time, used electrode sets, etc., are all followed by the procedures described section 4.4.1.

5.2.2 Classification

Two different approaches are applied to display the result of the classification: the static method where the result is displayed after a pre-determined number of sets of stimulus are flashed (i.e., after a fixed number of trial groups) and the threshold method where the result is provided when a threshold condition is satisfied, which is actually $(p_1 - p_2 - p_3) \geq 0.9$ where p_1, p_2, p_3 are respectively the first, second, and third best probability values satisfied by any three characters $s_1, s_2, s_3 \in S$. Since the scores for each row and column are needed to calculate the overall probability value of each character in the speller matrix, our classifier determines the most probable letter as soon as EEG scores of 12 epochs corresponding to each individual row and column are calculated. F, F-B, and Viterbi algorithms are employed during the online spelling. Each subject participated two spelling sessions in which one of these three algorithms is used in decision- making with either the static or the threshold method. The BLDA classifier and trigram letter probabilities are used in the classification algorithm and the number of stimulus repetitions (trial groups) is set to 5 for the static method. During online spelling, subjects ignore the errors and continue to type the next letter. Our overall online decision making algorithm follows the following procedure.

At the end of each trial group where an EEG score for each of the 12 individual row/column flashes are obtained, these EEG scores together with n-gram probabilities calculated from the text corpus are fed into the HMM and posterior probability scores of each letter was obtained by either Forward, Forward-backward, or Viterbi decoding. If these score distributions satisfy the threshold criterion given above for the threshold method, the most probable letter is displayed as the answer of the classification. For the static method, regardless of the threshold criteria, the result is just provided at the end of 5 stimulus repetitions. Therefore, in the threshold approach, the classifier has the opportunity to display the result without waiting 5 repetitions if the threshold criterion is satisfied. Decision for the current letter is made according to the Forward algorithm.

Later, for the F-B and Viterbi algorithms, the decision made in previous time instants can be updated after a decision in the current time is reached. Thus, there is a possibility to correct an error made in the previous time. This is not valid for the Forward algorithm since this is a recursive filtering algorithm operating on past and present data only, rather than future data. The resulted letter sequence which is obtained after updating the previous and current letters at the end of each time instant (run) is not displayed to the subject. First, current letter result is displayed to the subject and appended to Text box, simultaneously, the updating is processed and the new letter sequence is appended to Text box after deleting the previously written characters of a word. α, β, δ and emission probabilities obtained at the end of each run are stored to be used in next runs within each word. When the "_" character is spelled by the subject, the classifier realizes that there is a new word that the subject intends to spell. Then, the decoding algorithms are processed within the new word. For the online spelling experiments, we do not have a "backspace" character in the speller matrix. To let subjects use a "backspace" character could be considered as an option for correction of the errors made in each time instant if a wrong letter is decided by the algorithm. In this way, the subject would delete the wrong character by focusing on backspace, and then type again. However, to test the effectiveness of F-B and Viterbi methods to correct the errors by updating the previous decisions, we choose not to use the "backspace" character. By utilizing this character, subjects can spell the target letters with 100 % correct in real-time. However, the information rate of the system could be reduced due to the repetition of the wrongly spelled letters.

5.3 Results

Since the main of focus of the study described in this chapter is to develop and demonstrate the real-time implementation of our proposed method and, we did not perform the BLDA and NLP methods for the online spelling experiment. We believe that testing all 5 methods in five different sessions will take too much time of the subjects that and will lead to fatigue and loss of concentration during a session. Comparing the results of the F, F-B and Viterbi methods would also not provide healthy results due to the changes in the concentration of the subject across sessions.

Table 5.1 shows the result of online spelling for each subject using the F, F-B and Viterbi algorithms. On average, our online spelling system performs better accuracy when the sets of flashes are fixed to 5 compared with threshold approach. However, varying number of repetitions based on a threshold results in higher typing rate and bit-rate. If we neglect the time to show the classification result, the system with the threshold method can type 13.13 letters per minute. Moreover, average result of the static method is very consistent with the results in Table 4.7. Table 4.7 asserts that with the trigram language model, the proposed algorithm reaches accuracy value between 81% and 87% in the first 5 repetitions. In our online result, we achieve 83.7% average accuracy using 5 trial groups, which demonstrates the success of the system in real-time experiments. Table 5.1 asserts that several subjects perform better than the others. For example, Subject 1 and Subject 7 reach significant performance values, which demonstrates that our system can achieve both high accuracy and speed in terms of letters per minute, simultaneously, which was the main goal of this study as mentioned in the introduction chapter. To be more clear, these results conclude that using our system if we assume no delay between two consecutive runs, on average, a subject may type 20 letters in a minute with 91.7 percent correct that is actually pretty good value in P300 context. Our method adds a prior probability to the EEG scores obtained by the BLDA classifier based on the language model. This helps the system to satisfy the threshold condition more quickly by adding additional probabilistic information from the linguistic domain rather than presuming equal *a priori* probabilities for all characters. Because of this, the system can have the ability to achieve both high accuracy and speed by exploiting linguistic information. Of course, these results are subject-specific, however, the performances of subjects 1 and 7 show that this observation is valid for at least some subjects.

Table 5.1: Online Performances for each subject

	Static Method						**Threshold Method**					
	ALG	R	W	Acc	Rate	Bit-rate	ALG	R	W	Acc	Rate	Bit-rate
Subjects				(%)	(let./min)	(bits/min)				(%)	(l/m)	(bits/min)
S1	Vit	23	1	95.8	8	26.87	F-B	22	2	91.7	22.86	46.21
S2	-	-	-	-	-	-		16	8	66.7	9.97	16.93
	-	-	-	-	-	-	F	15	9	62.5	9.89	15.17
S3	F-B	23	2	92	8	24.90	-	-	-	-	-	-
	Vit	20	5	80	8	19.55	-	-	-	-	-	-
S4	F	23	1	95.8	8	26.87	-	-	-	-	-	-
	Vit	22	2	91.7	8	24.75	-	-	-	-	-	-
S5	-	-	-	-	-	-	F	16	9	64	11.49	17.37
	-	-	-	-	-	-	F-B	18	7	72	8.78	17.56
S6	F	12	13	48	8	8.60	F-B	15	10	60	9.40	13.73
S7	-	-	-	-	-	-	Vit	22	2	91.7	17.45	40.37
	-	-	-	-	-	-	F	19	5	79.2	15.22	29.09
Sum & Avg	All	123	24	**83.7**	**8**	**21.92**	All	143	52	**73.33**	**13.13**	**24.56**

Note: Vit:Viterbi Algorithm, ALG: Algorithm name, Acc: Accuracy, (l/m): letters/min,
R: # of correct answers, W: # of wrong answers.

Chapter 6

Conclusions and Future Work

In this thesis, we have presented a new P300-based BCI system incorporating a language model constructed by an HMM, and have demonstrated the performance of our proposed approach. In this direction, we have first introduced our proposed generative model utilizing both EEG scores obtained by two well performing classifiers and letter prior information obtained from a language domain with the use of a text corpus. We have then described filtering, smoothing, and Viterbi algorithms to combine these pieces of information within our generative model for inference. We have demonstrated how the uncertain but useful information from previous time instants is taken into account for the later time instants in a fully probabilistic approach and how both the past and feature information are incorporated to decide on the current letter. In this way, previously declared letters can be updated as new information arrives.

We have performed both offline and online spelling experiments with 7 healthy subjects. We utilized both Turkish and English languages for offline analysis of different datasets and have only used the Turkish language model for the real-time experiments performed in our own laboratory. Our results show that the proposed model utilizing language domain information can achieve higher speed and accuracy compared to relevant recent works. We have also shown that the impact of our language model is preserved when data from only a small number of electrodes are available. The results of our online decision- making experiment demonstrate the successful operation of the proposed approach in a real-time BCI setting.

Our results show that the overall performance improvement with the use of a trigram

language model is around 30% for accuracy and 60% for bit-rate compared to the method without utilizing a language model if limited date are available. If the order of the language model is increased, then the improvement is better. We also have shown that our results on the the offline analysis of the data outperform the results of a relevant piece of work utilizing a language model, namely the NLP method, where improvements of approximately 15 % for accuracy and 25 % for bit-rate are achieved by our approach. Furthermore, these improvements are also attained when the LR classifier is used and feature reduction is performed by reducing the number of available electrodes to make data worse which exactly demonstrates that the proposed model will exhibit robustness against NLP method to the potentially poor conditions of a data collection procedure. Our analysis on the BCI competition dataset shows that after several data pre-processing techniques are performed, we achieve 100 % accuracy in 3 stimulus repetitions on this dataset, which is the better than that obtained by the winner of the competition. As a final point, our main goal for online analysis was to implement a real-time P300 speller system which utilizes a language model to estimate the target letter. F, F-B, and Viterbi algorithms used in offline analysis were also implemented to be used in a real-time system and the results were reported. However, as we mentioned in Chapter 5, these results are represented just to show the capability of the system to reach satisfactory accuracy values with a high speed.

This study was performed to indicate the benefit of the use of a language model in BCI based typing. While the results are encouraging, we do not claim that this is the best model. In fact, we think that there are several future directions that can be developed to achieve better improvements both in offline and online analysis.

We anticipate that implementing a discriminative model such as Conditional Random Field (CRF) [77] where the relation between the letters, stimulus events and their corresponding EEG signals are represented with freely defined feature functions may eliminate the effect of the independence assumption on HMM for computing the observation data (EEG) letter scores and therefore, may exhibit better performance.

Another important aspect is the characterization of errors in classification. By analyzing the wrong answers in our experiments, we conclude that a letter adjacent to the

intended target is usually selected as the answer. Furthermore, an important portion of errors including letters irrelevant to the target are located in a far corner of the matrix. The effects that cause this type of wrong classification should be investigated in more detail. We believe that training a second classifier based on these errors should help in error-reduction or preventing erroneous feedback to the user [28].

A good future direction of this work can be adjusting the letter positions in the matrix interface according to letter estimations obtained from F, F-B and Viterbi algorithms. Using the EEG data up to present time instant, the classification algorithms can estimate the next letter and we can adjust the letter positions in the matrix based on the hypothesis that most target-error pairs lie on the same row or column. For F-B and Viterbi algorithms, the scores for each character in the matrix should be calculated using Forward algorithm and the letter positions can be adjusted according to that scores. In [71], a modified interface based on a custom-built dictionary was proposed. They achieve significant improvement in information transfer rate using an adaptive interface. However, the improvement can be even greater if a language model based approach is utilized. So adaptation of the structure of the spelling matrix within our framework could be an interesting line of future work.

More advanced models that better utilize knowledge of linguistic structure will likely provide greater improvements than the work presented here. For example, a simple improvement would be to include a model with word probabilities. The corpus used in this study contains part of speech tags which could provide additional prior information. Discourse and context information can also be integrated into this system. In addition to this, the corpus used in this study contains text samples from variety of domains and it is large enough to provide reliable n-gram counts. However, for the clinical implementation of this system, a different corpus that is more specific to the patient's needs may be more effective [15].

For the real-time implementation of this system, a word suggestion system can be added to the spelling software to perform dictionary lookups as characters are selected with the use of our method utilizing a language model. Several number of words can be sorted according their probabilities and if a target word is among the list, it may be selected by

the subject and the word completion is achieved without completing the remaining runs. Although this method is supposed to increase the speed of the system, in practice, one could face some limitations because of a more complicated graphical interface. In such a word suggestion system, characters are removed from their interface to make room for word completions and therefore the size of the matrix grid is diminished [15]. It could be tough for the subjects to focus a target letter in a grid where the characters are small and closely spaced. For example, the developed word suggestion system in [78] suffers from the lower accuracy although information transfer rate is increased. In short, the word suggestion systems are open to questions because of the trade-off between accuracy and speed for the real-time applications.

As we mention in Section 5.2.2, although we choose not to use a "backspace" character in online spelling experiments, it may be used for the decision-making algorithm based on Forward algorithm where previously written letters are not updated in future time. However, having the "backspace" character as an element of the speller matrix can be problematic due to the calculation of a conditional probability value for this character. We can neither assign a very low probability value for "backspace" as we did for the number characters nor learn the conditional letter likelihood for this character using a text corpus as we did for letter characters of the matrix. A new model may need to be developed to approximate the likelihood of making error for each subject. However, this issue is still open to questions and should be thought about in more detail.

We should also note the performance of our online algorithm is not at its best because of the fact that the threshold criterion was fixed for all subjects. We believe that once a learning algorithm is developed for detecting optimum confidence threshold value for subjects based on their training data, the performance would be much better. In other words, performance could be even greater by subject-adaptive selection of the threshold.

To sum up, the P300 speller is still an attractive field of research that has many unanswered questions. We are planning to address some of the issues mentioned here in more detail in the near future.

Bibliography

[1] Lu J, Speier W, Hu X, and Pouratian N, " The effects of stimulus timing features on P300 speller performance," *Clinical Neurophysiology*, vol. 124(2), pp. 306-14, 2013.

[2] Guger C, Daban S, Sellers E, Holzner C, Krausz G, Carabalona R, Gramatica F, and Edlinger G, " How many people are able to control a p300-based brain-computer interface (bci)?," *Neurosci. Lett.*, vol. 462(1), pp. 94-98, 2009.

[3] Cohen D, " Magnetoencephalography: detection of the brain's electrical activity with a superconducting magnetometer," *Science*, vol. 175, pp. 664-66, 1972.

[4] Sitaram R, Caria A, Veit R, Gaber T, Rota G, Kuebler A, and Birbaumer N, " fmri brain-computer interface: a tool for neuroscienti fic research and treatment," *Intell. Neuroscience*, vol. 1, pp.1-10, 2007.

[5] Harmany Z, Marcia R, and Willett R, " SPIRAL out of Convexity: Sparsity-regularized Algorithms for Photon-limited Imaging, " *SPIE Electronic Imaging*, 2010.

[6] Coyle S. M., Ward T. S. E., and Markham C. M., " Brain–computer interface using a simplified functional near-infrared spectroscopy system,"*Journal of Neural Engineering*, vol. 4 (3), pp. 219–226, 2007.

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

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

[9] Lauer R, Peckham P, Kilgor K, and Heetderks W, " Applications of cortical signals to neuroprosthetic control: a critical review," *IEEE Trans. Rehabil. Eng*, vol. 8(2), pp. 205-8, 2000.

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

[11] Sellers E and Donchin E, " A P300-based brain-computer interface: initial tests by ALS patients, " *Clin Neurophysiol*, vol. 117, pp. 538-48, 2006.

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

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

[14] Krusienski D, Sellers E, McFarland D, Vaughan T, and Wolpaw J, " Toward Enhanced P300 Speller Performance," *J Neurosci Methods*,vol. 167(1), pp. 15-21, 2008.

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

[16] Orhan U, Erdogmus D, Roark B, Oken B, Purwar S, Hild K, Fowler A, and Fried-Oken M, " 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, CA), 2012.

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

[18] Kaper M, Meinicke P, Grossekathoefer U, Lingner T, and Ritter H, " BCI competition 2003-data set IIB: Support vector machines for the P300 speller paradigm," *IEEE Trans. Biomed. Eng.*, vol.51, pp. 1073-1076, 2004.

[19] Krusienski D, Sellers E, Cabestaing F, Bayoudh S, Mc- Farland D, Vaughan T, and Wolpaw J, " A comparison of classification techniques for the P300 speller," *J. Neural Eng.*, vol.3, pp.299-305, 2006.

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

[21] Allison B and Pineda J, " ERPS evoked by different matrix sizes:Implications for a BCI system," *IEEE Trans. Neural Syst. Rehabil. Eng.* vol.11, pp. 110–113, 2003.

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

[23] McFarland D, Sarnacki W, Townsend G, Vaughan T, and Wolpaw J, " The P300-based brain–computer interface (BCI): effects of stimulus rate," *Clin. Neurophysiol.*, vol.122, pp. 731–7, 2011.

[24] Martens S, Mooij J, Hill N, Farquhar J, and Schölkopf B, " A graphical model framework for decoding in the visual ERP-based BCI speller," *Neural Computation*, vol. 23, pp. 160-82, 2011.

[25] Kindermans P-J, Verschore H, Verstraeten D, and Schrauwen B, " A P300 BCI for the masses: prior information enables instant unsupervised spelling," *Advances in Neural Information Processing Systems 25*, 2012.

[26] Ulas C and Cetin M, " Incorporation of a language model into a Brain-Computer Interface based speller through HMM," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Vancouver-Canada), 2013.

[27] Verschore H, " A Brain-Computer Interface combined with a language model: the requirements and benefits of a P300 speller," *Master Thesis, Universiteit Gent*, 2012.

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

[29] Biosemi Headcaps, `http://www.biosemi.com/headcap.htm`.

[30] Biosemi Active electrodes, `http://www.biosemi.com/active_electrode.htm`.

[31] Niedermeyer E and Lopes da Silva F, " EEG Recording and Operation of the Apparatus," *in Electroencephalography : basic principles, clinical applications, and related fields*, 5th ed Philadelphia ; London: Lippincott Williams & Wilkins, pp.140-141, 2005.

[32] Biosemi Active Electrodes, `http://www.biosemi.com/pin_electrode.htm`.

[33] Wang M, " Design of a Modified P300 Speller System Based on Prediction by Partial Matching Language Model," *Master Thesis,University of Cincinnati*, 2012.

[34] BCI system diagram, `http://www.dewaldlab.com/research/BCI.htm`.

[35] Müller-Putz G.R et al., " Steady-state visual evoked potential (SSVEP)-based communication: impact of harmonic frequency components, " *J. Neural Eng.*, vol. 2 123, 2005.

[36] Hinterberger T et al., " Brain-computer communication and slow cortical potentials, " *IEEE Trans Biomed Eng.*, vol. 51(6), 1011-8, 2004.

[37] Pfurtscheller G, Neuper Ch, Flotzinger D, and Prenzenger M, " EEG-based discrimination between imagination of right and left hand movement," *Electroenceph.and Cli. Neurophys.*, vol.103, pp. 642-51, 1997.

[38] Celesia G and Brigell M, " Event-related potentials," *Curr Opin Neurol Neurosurg.*, vol. 5(5), pp. 733-9, 1992.

[39] Donchin E and Smith DBD, " The Contingent Negative Variation and the Late Positive Wave of the Average Evoked Potential," *Electroencephalogr Clin Neurophysiol*, vol. 29, pp. 201-203, 1970.

[40] Kübler A and Müller K, " An Introduction to Brain-Computer Interfacing," *in Toward Brain-Computer Interfacing*, G. Dornhege, et al., Eds., ed: MIT Press, pp. 1-25, 2007.

[41] Picton T.W, " The P300 wave of the human event-related potential," *Journal of clinical neurophysiology*, vol. 9(4), pp. 456-79, 1992.

[42] Squires N, Squires K, and Hillyard S, " Two varieties of long latency positive waves evoked by unpredictable auditory stimuli in man," *Electroenceph. and clin Neurophysi.*, vol. 38(4), pp. 387-401, 1975.

[43] Blankertz B, Krauledat M, Dornhege G, Williamson J, Murray-Smith R, and Müller K-R, " A note on brain actuated spelling with the Berlin brain-computer interface," *Proceedings of the 4th international conference on Universal access in human-computer interaction: ambient interaction*, pp.759-68, 2007.

[44] Hild K, Orhan U, Erdogmus D Roark B, Oken B, " Purwar S, Nezamfar H, and Fried-Oken M, " An ERP-based Brain-Computer Interface for text entry using Rapid Serial Visual Presentation and Language Modeling," *Proceedings of the ACL-HLT 2011 System Demonstrations*, pp.38-43, 2011.

[45] Schalk G, McFarland D.J, Hinterberger T, Birbaumer N, and Wolpaw J.R, " BCI2000: A General-Purpose Brain-Computer Interface (BCI) System, " . *IEEE Trans Biomed Eng.*, vol. 51(6), June 2004.

[46] BCI 2000 Speller, `http://www.bci2000.org/wiki/index.php/User_Reference:P3SpellerTask`.

[47] Hoffmann U, Vesin JM, Ebrahimi T, and Diserens K, " An efficient P300-based brain-computer interface for disabled subjects, " *J Neurosci. Methods*, vol. 167, pp. 115-25, 2008.

[48] Selim A.E, Wahed M.A, and Kadah Y.M, " Machine Learning Methodologies in Brain-Computer Interface Systems," *Biomedical Engineering Conference, CIBEC* , pp 1-5, 2008.

[49] Johnson G and Krusienski D, " Ensemble SWLDA Classifiers for the P300 Speller," *Proceedings of the 13th International Conference on Human-Computer Interaction. Part II: Novel Interaction Methods and Techniques* , pp. 551-57, 2009.

[50] Donchin E, Spencer K.M, and Wijesinghe R, " The mental prosthesis: assessing the speed of a P300-based brain-computer interface," *IEEE Trans Rehabil Eng*, vol. 8, pp. 174-9, 2000.

[51] Bishop C.M, " Pattern Recognition and Machine Learning," *Springer*, 2006.

[52] Thulasidas M and Guan C, " Optimization of BCI Speller Based on P300 Potential," *Conf Proc IEEE Eng Med Biol Soc*, vol. 5, pp. 5396-9, 2005.

[53] Hiremath P.S and Tegnoor J.R, *Follicle Detection and Ovarian Classification in Digital Ultrasound Images of Ovaries, Advancements and Breakthroughs in Ultrasound Imaging,* Dr. G P P Gunarathne (Ed.), ISBN: 978-953-51-1159-7, InTech, DOI: 10.5772/56518, 2013. Available from: http://www.intechopen.com/books/advancements-and-breakthroughs-in-ultrasound-imaging/follicle-detection-and-ovarian-classification-in-digital-ultrasound-images-of-ovaries.

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

[55] Jin J, Sellers E, and Wang X, " Targeting an efcient target-to-target interval for P300 speller brain–computer interfaces, *Med Biol Eng Comput.*, vol. 50, pp. 289-96, 2012.

[56] Amcalar A and Cetin M, " Design, implementation and evaluation of a real-time P300-based brain-computer interface system," *International Conference on Pattern Recognition (ICPR)*, pp. 117-20, 2010.

[57] Li K, Sankar R, Arbel Y, and Donchin E, " Single trial independent component analysis for P300 BCI system," *Conf Proc IEEE Eng Med Biol Soc (EMBC)*, pp. 4035-8, 2009.

[58] Makeig S, Jung T-P, Bell A, Ghahremani D, and Sejnowski T, " Blind separation of auditory event-related brain responses into independentcomponents," *Proc Natl Acad Sci USA*, vol. 94, pp. 10979-84, 1997.

[59] Rosenfeld R, " A maximum entropy approach to adaptive statistical language modelling, " *Computer speech and language*, vol. 10(3), pp. 187-228, 1996.

[60] Mitchell T, " Machine Learning," *McGraw Hill*, A draft version : `http://www.cs.cmu.edu/~tom/mlbook/NBayesLogReg.pdf`

[61] Rabiner L.R, " A Tutorial on hidden Markov models and selected applications in speech recognition, " *Proceedings of the IEEE*, vol. 77(1), 257-286, 1989.

[62] Rabiner L.R and Juang B.H, " An introduction to hidden Markov models, " *IEEE ASSP Magazine*, pp. 4–15, 1986.

[63] Massey F, " The Kolmogorov–Smirnov test for goodness of fit, " *J. Am. Stat. Assoc.*, vol. 46, pp. 68–78, 1957.

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

[65] Du Preez J.A, " Efficient High-Order Hidden Markov Modelling, " *Doctor of Philosophy Dissertation, University of Stellenbosch*, 1997.

[66] Manning C and Schütze H, " Foundations of Statistical Natural Language Processing," *MIT Press* (Cambridge, MA), 1999.

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

[68] Ververidis D and Kotropoulos C, " Fast and accurate sequential floating forward feature selection with the Bayes classifier applied to speech emotion recognition, " *Signal Processing*, vol. 88, pp. 2956-70, 2008.

[69] Pudil P, Novovicova J, and Kittler J, " Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, pp. 1119-1125, 1994.

[70] Blankertz B, BCI Competition II webpage. [Online]. Available: `http://www.bbci.de/competition/ii/`.

[71] Ahi S, Kambara H, and Koike Y, " A dictionary-driven P300 Speller with a modified interface," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 19, pp. 6-14, 2011.

[72] Beresford R, The UK Advanced Cryptics Dictionary [Online]. Available : `http://cfaj.freeshell.org/wordfinder/UKACD17.shtml`.

[73] Alhaddad M.J, " Common Average Reference (CAR) Improves P300 Speller," *International Journal of Engineering and Technology*, vol. 2(3), 451-64, 2012.

[74] Xu N, Gao X, Hong B, Miao X, Gao S, and Yang F, " BCI competition 2003-data set IIb: enhancing P300 wave detection using ICA-based subspace projections for BCI applications," *IEEE Trans. Biomed. Engineering*, vol. 51(6), pp. 1067-1072, 2004.

[75] Bostanov V, " BCI competition 2003-data sets Ib and IIb: feature extraction from event-related brain potentials with the continuous wavelet transform and the t-value scalogram," *IEEE Trans. Biomed. Engineering*, vol. 51(6), pp. 1057-1061, 2004.

[76] Blankertz B, et al., " The BCI Competition 2003: progress and perspectives in detection and discrimination of EEG single trials," *IEEE Trans Biomed Eng*, vol. 51, pp. 1044-51, 2004.

[77] Lafferty J, McCallum A, and Pereira F, " Conditional random fields: Probabilistic models for segmenting and labeling sequence data," *Proc. 18th International Conf. on Machine Learning*, Morgan Kaufmann, pp. 282–89, 2001.

[78] Ryan D, Frye G, Townsend G, Berry D, Mesa-G S, Gates N, and Sellers E, " Predictive spelling with a P300-based brain–computer interface: increasing the rate of communication," *Int. J. Hum.–Comput. Interact.*, vol.27, pp. 69–84, 2011.