

UCLA

UCLA Previously Published Works

Title

Integrating Language Information With a Hidden Markov Model to Improve Communication Rate in the P300 Speller

Permalink

<https://escholarship.org/uc/item/1qb7953c>

Journal

IEEE Transactions on Neural Systems and Rehabilitation Engineering, 22(3)

ISSN

1534-4320

Authors

Speier, William
Arnold, Corey
Lu, Jessica
[et al.](#)

Publication Date

2014-05-01

DOI

10.1109/tnsre.2014.2300091

Peer reviewed



Published in final edited form as:

IEEE Trans Neural Syst Rehabil Eng. 2014 May ; 22(3): 678–684. doi:10.1109/TNSRE.2014.2300091.

Integrating language information with a hidden Markov model to improve communication rate in the P300 speller

William Speier,

Bioengineering Department and the Medical Imaging Informatics Group, University of California, Los Angeles, CA 90024 USA

Corey Arnold,

Medical Imaging Informatics Group, University of California, Los Angeles, CA 90024 USA

Jessica Lu,

Bioengineering Department, University of California, Los Angeles, CA 90024 USA

Aniket Deshpande, and

Bioengineering Department, University of California, Los Angeles, CA 90024 USA

Nader Pouratian

Bioengineering and Neurosurgery departments, the Neuroscience Interdepartmental Program, and the Brain Research Institute, University of California, Los Angeles, CA 90024 USA

William Speier: Speier@ucla.edu; Corey Arnold: cwarnold@ucla.edu; Jessica Lu: jessicalu@ucla.edu; Aniket Deshpande: aniket@ucla.edu; Nader Pouratian: NPouratian@mednet.ucla.edu

Abstract

The P300 speller is a common brain–computer interface (BCI) application designed to communicate language by detecting event related potentials in a subject’s electroencephalogram (EEG) signal. Information about the structure of natural language can be valuable for BCI communication systems, but few attempts have been made to incorporate this domain knowledge into the classifier. In this study, we treat BCI communication as a hidden Markov model (HMM) where hidden states are target characters and the EEG signal is the visible output. Using the Viterbi algorithm, language information can be incorporated in classification and errors can be corrected automatically. This method was first evaluated offline on a dataset of 15 healthy subjects who had a significant increase in bit rate from a previously published naïve Bayes approach and an average 32% increase from standard classification with dynamic stopping. An online pilot study of five healthy subjects verified these results as the average bit rate achieved using the HMM method was significantly higher than that using the naïve Bayes and standard methods. These findings strongly support the integration of domain-specific knowledge into BCI classification to improve system performance and accuracy.

Index Terms

Brain-computer interfaces; Natural Language Processing; Electroencephalography; Hidden Markov Models

I. Introduction

High brain stem injuries and motor neuron diseases such as amyotrophic lateral sclerosis (ALS) can interrupt the transmission of signals from the central nervous system to effector muscles, impairing a patient's ability to communicate and causing them to become 'locked-in'. Brain-computer interfaces (BCI) restore some of this ability by detecting electrical signals from the brain and translating them into computer commands [1]. The P300 Speller is a common BCI system that uses electroencephalogram (EEG) signals to simulate keyboard input, providing a means of communication [2].

The P300 speller works by presenting a grid of characters on a graphical interface and instructing the user to focus on a target letter. Characters are illuminated (i.e., "flashed") in a pseudo-random manner and evoked responses known as P300 signals are elicited when the target character is flashed, which are detected by a classifier. Because the signal to noise ratio is low, several trials must be combined in order to correctly classify responses. The resulting typing speed can therefore be slow, prompting many studies focused on system optimization. Approaches that have been adopted to accomplish such optimization include varying the grid size [3–5], optimizing system parameters [6,7], and adopting different signal classification algorithms [8–11].

While the P300 speller is designed to provide a means for communication, most attempts at system optimization have not taken advantage of existing knowledge about the language domain. Most existing analyses treat character selections as independent elements chosen from a set with no prior information, but information about the domain of natural language can be used to create a prior belief about the characters to be chosen. It has been shown that system speed and accuracy can be improved when adding an n-gram language model as a prior for naïve Bayes [12], partially observable Markov decision process [13], and expectation maximization [14] algorithms.

Automatic error correction is another technique that has been largely unexplored in BCI communication. Traditionally, users have been instructed to correct errors as they occur, but in some cases such corrections may be unnecessary with an automatic correction technique. Many non-BCI typing methods employ automatic correction, including programs for word processing [15] and text messaging [16]. The field of BCI has witnessed some movement in this direction, as illustrated in a system by Ryan et al. whereby subjects can ignore errors when they are typed and correct them later with suggested words from a dictionary [17].

In this work, we build upon our previous system by modeling typing with the P300 speller as a hidden Markov model (HMM). An HMM treats typing as a sequential process where each character selection is influenced by the previous selection. The model is hidden because we cannot observe user intent directly. Instead, we use the Viterbi algorithm to determine the optimal sequence of target characters given the observed EEG signal. This method was compared offline with a standard stepwise linear discriminant analysis (SWLDA) method with dynamic stopping as well as a previously published naïve Bayes classifier (NB) [12] on a set of 15 healthy subjects. These results were then verified through a five subject online pilot study.

II. Materials and methods

A. Data Collection

The subjects in the offline dataset were 15 healthy graduate students and faculty with normal or corrected to normal vision between the ages of 20 and 35. Only one subject (subject F) had previous experience using a BCI for typing. The system used a 6×6 character grid, row and column flashes, and an interstimulus interval (ISI) of 125 ms. Each subject underwent between 8 and 10 trials consisting of spelling a five letter word (Table 1) with 15 sets of 12 flashes (six rows and six columns) for each letter. The choice of target words for this experiment was independent of the trigram language model used in the NB and HMM methods.

The subjects for the online study consisted of five healthy volunteers with normal or corrected to normal vision between the ages of 23 and 30. The training sessions for these subjects consisted of three sessions of copy spelling 10 character phrases. Each subject then chose a target phrase to spell in online sessions. In each session, the subject had five minutes to spell as much of the phrase as they could using one of the three analysis methods: SWLDA (without dynamic stopping), NB, or HMM. Subjects were instructed not to correct errors and to repeat the phrase if they completed it in under five minutes. Two different threshold values were used for each of the analysis methods (5 and 10 sets of flashes for SWLDA, and .95 and .98 for NB and HMM) for a total of six sessions. The threshold that yielded the best value for each method was chosen as the optimal value and the corresponding output was used to determine the subject's performance.

BCI2000 was used for data acquisition and analysis was performed offline using MATLAB (version 7.10.0, MathWorks, Inc, Natick, MA). Three analysis methods were compared using stepwise linear discriminant analysis, naïve Bayes, and hidden Markov models, respectively.

B. SWLDA

SWLDA is a classification algorithm that selects a set of signal features to include in a discriminant function [18]. Training was performed using cross-validation between trials where the test set was one of the trial words and the rest were the training set.

The signals in the training set were assigned labels based on two classes: those corresponding to flashes containing the attended character and those without the attended character. Each new signal was then reduced to a score that reflects how similar it was to the attended class.

The algorithm used ordinary least-squares regression to predict class labels for the training set. It then added the features that were most significant in the forward stepwise analysis and removed the least significant features in the backward analysis step. These steps were repeated until either the target number of features was met or it reached a state where no features were added or removed [11].

The score for each flash in the training set, y_t^i , was then computed as the dot product of the feature weight vector, \mathbf{w} , with the features from that trial's signal, \mathbf{z}_t^i :

$$y_t^i = \mathbf{w} \cdot \mathbf{z}_t^i \quad (1)$$

Scores for each flash in the training set were then computed and the distributions for the attended and non-attended signals were found. We assumed that scores were independent and normally distributed, resulting in the probability distributions

$$f(y_t^i | x_t) = \begin{cases} \frac{1}{\sqrt{2\pi\sigma_a^2}} e^{-\frac{1}{2\sigma_a^2}(y_t^i - \mu_a)^2} & \text{if } x_t \in A_t^i \\ \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{1}{2\sigma_n^2}(y_t^i - \mu_n)^2} & \text{if } x_t \notin A_t^i \end{cases} \quad (2)$$

where A_t^i is the set of characters illuminated for the i th flash for character t in the sequence and μ_a, σ_a^2, μ_n and σ_n^2 are the means and variances of the distributions for the attended and non-attended flashes, respectively. Each stimulus response is assumed to be independent, resulting in the conditional distribution

$$p(\mathbf{y}_t | x_t) \propto \prod_i f(y_t^i | x_t) \quad (3)$$

A threshold probability, p_{Thresh} , was then set to determine when a decision should be made. The program flashed characters until either $\max_{x_t} p(\mathbf{y}_t | x_t) > p_{Thresh}$ or the number of sets of flashes reached the maximum (15). The classifier then selected the character that satisfied $\text{argmax}_{x_t} p(\mathbf{y}_t | x_t)$. The speeds, accuracies, and bit rates were found for values of p_{Thresh} between 0 and 1 in increments of 0.01 and the threshold probability that maximized the bit rate was chosen for each subject.

C. NB

The NB method uses conditional distributions, $p(\mathbf{y}_t | x_t)$, found in the same way as in the SWLDA method. A naïve Bayes classifier was used to determine the probability of each character given the flash scores and the previous decisions [19]. If we assume that the individual flashes are conditionally independent given the current attended character, the posterior probability is

$$p(x_t | \mathbf{y}_t, x_{t-1}, \dots, x_0) = \frac{p(x_t | x_{t-1}, \dots, x_0) p(\mathbf{y}_t | x_t, \dots, x_0)}{p(\mathbf{y}_t | x_{t-1}, \dots, x_0)} \propto p(x_t | x_{t-1}, \dots, x_0) \prod_i f(y_t^i | x_t) \quad (4)$$

Prior probabilities for characters were obtained from frequency statistics in an English language corpus [20]. This probability was simplified using the second-order Markov assumption to create a trigram model [21]. The prior probability that the next character is x_t given that the last two characters chosen were x_{t-1} and x_{t-2} is then equal to the number of

times that all three characters occurred together in the corpus divided by the number of times the last two characters occurred together:

$$p(x_t|x_{t-1}, \dots, x_0) = \frac{c(x_{t-2}, x_{t-1}, x_t)}{\sum_{x_t} c(x_{t-2}, x_{t-1}, x_t)} = \frac{c(x_{t-2}, x_{t-1}, x_t)}{c(x_{t-2}, x_{t-1})} \quad (5)$$

where $c(x_{t-2}, x_{t-1}, x_t)$ is the number of occurrences of the string ' $x_{t-2}x_{t-1}x_t$ ' in the corpus. Trigrams for the English language were obtained from the Brown corpus, a compilation of over 2 million English documents published in the United States in 1961 [20].

Similar to the SWLDA method, a threshold probability, p_{Thresh} , was then set to determine when a decision should be made. The program flashed characters until either $\max_{x_t} p(x_t|y_t, x_{t-1}, \dots, x_0) > p_{Thresh}$ or the number of sets of flashes reached the maximum (15). The classifier then selected the character that satisfied $\operatorname{argmax}_{x_t} p(x_t|y_t, x_{t-1}, \dots, x_0)$. Again, the speeds, accuracies, and bit rates were found for values of p_{Thresh} between 0 and 1 in increments of 0.01 and the threshold probability that maximized the bit rate was chosen for each subject.

D. HMM

Hidden Markov models are used to model Markov processes that cannot be directly observed, but can be indirectly estimated by state-dependent output. The goal of such systems is to determine the optimal sequence of states in the Markov process that could have produced an observed output sequence.

The HMM method treats typing as an n th order Markov process. States in the process consist of tuples representing the target character and the previous $n-1$ targets, $\mathbf{x}_t = \langle x_t, \dots, x_{t-n+1} \rangle$. Transition probabilities correspond to the conditional probability of the next state, \mathbf{x}_t , given the previous state, \mathbf{x}'_{t-1} ,

$$p(\mathbf{x}_t|\mathbf{x}'_{t-1}) = \begin{cases} p(x_t|x_{t-1}, \dots, x_{t-n}) & x_i = x'_{i-1}, \forall i \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

A typed word is then simply a sequence of states of the Markov process, $\mathbf{x} = (\mathbf{x}_0, \dots, \mathbf{x}_n)$. Because we cannot directly inspect the states of the process, we observe indirectly through the EEG signals. The EEG response is dependent only on the current state and governed by the conditional probability, $p(\mathbf{y}_t|x_t)$, which is defined as in the NB method. The goal is to determine \mathbf{x} through observation of the EEG signals, $\mathbf{Y} = (\mathbf{y}_0, \dots, \mathbf{y}_n)$

At each time point, t , the probability of the current state is computed using the forward step of the forward-backward algorithm:

$$p(\mathbf{x}_t|\mathbf{y}_t, \dots, \mathbf{y}_0) \propto p(\mathbf{y}_t|x_t) \sum_{\mathbf{x}'_{t-1}} p(\mathbf{x}_t|\mathbf{x}'_{t-1}) p(\mathbf{x}'_{t-1}|\mathbf{y}_{t-1}, \dots, \mathbf{y}_0) \quad (7)$$

As in the NB method, a selection occurs when the probability of a target character exceeds a threshold probability, p_{Thresh} . This is found by summing over all of the states that share the same character at time t .

$$\max_{x_t} p(x_t | \mathbf{y}_t, \dots, \mathbf{y}_0) = \max_{x_t} \sum_{x_{t-1}, \dots, x_{t-n+1}} p(x_t | \mathbf{y}_t, \dots, \mathbf{y}_0) \geq p_{Thresh} \quad (8)$$

At each time step, the Viterbi algorithm is used to determine the path to each state with the highest probability.

$$V_t(\mathbf{x}_t) = \max_{\mathbf{x}'_{t-1}} p(\mathbf{y}_t | x_t) p(x_t | \mathbf{x}'_{t-1}) V_{t-1}(\mathbf{x}'_{t-1}) \quad (9)$$

Back pointers are saved so that the optimal sequence ending in that state can be retrieved (Fig. 1). For each state, \mathbf{x}_t , a pointer is created to the state which satisfies

$$\operatorname{argmax}_{\mathbf{x}'_{t-1}} p(\mathbf{y}_t | x_t) p(x_t | \mathbf{x}'_{t-1}) V_{t-1}(\mathbf{x}'_{t-1}) \quad (10)$$

The optimal state for the current time step, \mathbf{x}_t , is selected such that it satisfies

$$\operatorname{argmax}_{\mathbf{x}_t} p(\mathbf{x}_t | \mathbf{y}_t, \dots, \mathbf{y}_0) \quad (11)$$

The back pointers are then followed from the selected state to find the optimal sequence of states up to the current time step. The corresponding sequence of characters is considered the most probable string typed. Each time a selection is made, the entire sequence is overwritten by the current optimal string. In most cases, the sequence \mathbf{x}_{t-1} has significant overlap with \mathbf{x}_t , so the new string is simply the old string with an appended character and possibly some corrected errors. For example, at a given time t , the state x_t may have the highest probability. However, at time $t+1$, state x_{t+1} may have the highest probability with an optimal transition from x'_t . The system would then go back and change the previous character to x'_t (Fig. 1).

This study used a second order Markov process (i.e., $n = 2$) for this method to stay consistent with the language model used by the NB method. The states of the model are then $\mathbf{x}_t = \langle x_t, x_{t-n+1} \rangle$ and the transition probabilities are the conditional probabilities $p(x_t | x_{t-1}, x_{t-2})$. As in the previous methods, the speeds, accuracies, and bit rates were found for values of p_{Thresh} between 0 and 1 in increments of 0.01 and the threshold probability that maximized the bit rate was chosen for each subject.

E. Evaluation

Evaluation of a BCI system must take into account two factors: the ability of the system to achieve the desired result and the amount of time required to reach that result. The efficacy of the system can be measured as the selection accuracy, which we evaluated by dividing the number of correct selections by the total number of trials.

For each model we also calculated the selection rate (SR). First, the average amount of time for a selection is found by adding the gap between flashes (3.5 s) to the product of the amount of time required for a flash (.125 s), the average number of sets of flashes (\bar{s}) and the number of flashes in each set (12). The selection rate measured in selections per minute is then the inverse of the average selection time:

$$SR = \frac{60}{3.5 + 0.125 * 12 * \bar{s}} \quad (12)$$

As there is a tradeoff between speed and accuracy, we also use bit rate as a metric, which takes both into account. The bits per symbol, B , is a measure of how much information is transmitted in a selection taking into account the accuracy and the number of possible selections [22]:

$$B = \log_2 N + P \log_2 P + (1-P) \log_2 \frac{1-P}{N-1} \quad (13)$$

where N is the number of characters in the grid (36) and P is the selection accuracy. The information transfer rate (ITR) (in bits min^{-1}) can then be found by multiplying the selection rate by the bits per symbol. Significance was tested using paired two-sample t -tests with 14 degrees of freedom.

Although the number of flashes was fixed for all trials, different selection rates were simulated by limiting the amount of data available for the classification algorithm. For example, if the confidence threshold is reached after 100 flashes, the corresponding data is used for classification and the rest is omitted.

III. Results

A. Offline

When using SWLDA in offline analysis, all subjects were able to type with varying levels of performance. The best performer (subject D) was able to achieve 89% accuracy at a rate of 9.96 selections per minute, while the worst performer (subject C) achieved an accuracy of 80% at a rate of only 3.96 selections per minute. The accuracy increased with the number of flashes for all subjects and 12 of the 15 were able to exceed 90% accuracy within 15 sets of flashes.

The optimal number of sets of flashes varied from 3 to 8, which yielded bit rates from 13.54 to 40.82 (Table 2). In general, subjects that performed better achieved an optimal bit rate in fewer flashes. On average, the subjects had a 29% accuracy after a single set of flashes which increased to about 95% after 15 sets. The average selection rate for the static method was 5.87, the average accuracy was 88.82%, and the average bit rate was 24.44.

The maximum bit rates using the naïve Bayes classifier improved by 50% on average ($p < 10^{-8}$), ranging from 39% (subject E) to 65% (subject C) compared to the SWLDA results (Table 2). The selection rate rose significantly ($p = 0.0002$) which the accuracy remained relatively constant ($p = 0.5$). In some cases, a decreased accuracy (subject D) or selection

rate (subject B) was reported for the NB method relative to the SWLDA method because of the optimization based on bit rate. In these cases, the optimal bit rate may occur at a lower accuracy, but a much higher selection rate, or vice versa.

Six of the subjects reached 100% accuracy within the 15 sets of flashes using the HMM method and subject D had all characters correct within two sets of flashes. The improvement in ITR from the static method to the HMM method ranged from 9% (subject N) to 85% (subject I). The average bit rate across subjects improved by 32% from 24.44 to 32.33 ($p < 10^{-8}$). The selection rate rose from 5.87 to 7.88 ($p < 10^{-5}$) and the accuracy increased stayed relatively constant ($p = 0.35$). The HMM method had a significantly higher ITR than the naïve Bayes method ($p=0.001$), although the optimal selection rates and accuracies were not significantly different ($p=0.09$ and $p=0.43$ respectively).

B. Online

In the online experiments, all five subjects were able to select characters with at least 75% accuracy using each of the methods (Table 3). Using the SWLDA method, four of the five subjects performed better using the five flash set, resulting in an average selection rate of 5.07, an average accuracy of 91.68%, and an average ITR of 22.35. Two of the five subjects achieved 100% accuracy with both thresholds tested. Using the NB classifier, the average accuracy dropped to 82.83%, but the selection rate rose significantly to 9.35 ($p=0.003$), resulting in a significantly high ITR (33.80, $p=0.0004$).

Four of the five subjects performed best using the HMM classifier. The average selection rate was 9.55 characters/minute with an accuracy of 92.34%, resulting in an average bit rate of 42.31, which was significantly higher than those achieved using SWLDA ($p=0.0003$) and NB ($p=0.02$). On average, the HMM method was able to correct 3 errors per person, which accounted for 47% of the total errors made using this method (Table 4).

IV. Discussion

Most current BCI communication systems do not utilize domain knowledge when processing natural language. Using a hidden Markov model to integrate this information can significantly improve the results of such a system.

In offline analysis, the SWLDA method achieved an average bit rate of 20.07 across subjects before implementing dynamic stopping, which is consistent with previous studies [4, 17]. Dynamic stopping improved this value 22% on average, which is close to previously reported values [10, 12]. Our HMM method improves the bit rate by an 32% on average to 32.33 relative to SWLDA. The offline improvement over the naïve Bayes method is more modest (7.1%), but still statistically and likely clinically significant ($p=0.001$). It should be appreciated that any improvement in performance can be considered useful since these BCI are intended to be the primary modality of communication for affected patients.

In general, the offline improvements over naïve Bayes seen after optimization are in speed rather than accuracy, which may be counterintuitive. This occurs because the HMM method has a stronger prior probability because it utilizes all of the past information. It is therefore

able to reach a similar accuracy with a lower confidence threshold, resulting in faster typing speed. When the confidence threshold is held constant in online tests, the typing speed is almost identical, but the accuracy increases due to the improved classifier.

In online analysis, subjects achieved an average ITR of 22.35 when using the SWLDA method. Two of the subjects achieved perfect accuracy using both configurations, indicating the system was not optimized for them as the minimal number of flashes for consistent performance was not reached. Despite the lack of a systematic optimization, the average performance online was superior to the offline performance. Similarly, both the NB and HMM methods demonstrated large improvements in online performance over offline. This could have been due to motivation from receiving feedback or because of randomness due to a small sample size.

The online performance increase of HMM over NB was much larger than that observed in offline analysis (25% for online, 7% for offline). This is likely attributable to the fact that, without error correction, errors compound in the NB method. Drastic errors such as missing a space will change the prior for subsequent characters, resulting in additional errors (Table 4). In the HMM method, these errors are corrected, while less significant errors such as switching vowels are often missed. It may be possible to identify which errors are more likely to be caught by the HMM method, allowing users to trust the system to account for some errors while manually correcting those that might be missed. Allowing manual error correction could lead to improved results in one or both of these methods (see future directions).

The ability to retrospectively change prior miscategorized letters automatically could have significant benefits to users. However, some users could also be distracted or discouraged by the presence of an incorrect selection and feel compelled to manually correct it. Similar to other non-BCI typing methods, we believe that a user could learn to trust the system and adapt to a modified task that did not involve correcting errors, which is supported by the success of the subjects in our online study. We also note that the proposed HMM method remains compatible with a backspace option, and if a user is inclined to correct all errors manually, the system reduces to the NB method and is still an improvement over SWLDA.

A. Limitations and future directions

A standard system and classification method were used in this study, but NLP could be used to integrate domain knowledge into any BCI communication system. Also, the Viterbi method and language model prior can be combined with any classifier that returns a likelihood probability. Studying the effects of integrating NLP into other BCI systems and in conjunction with different analysis methods remains as future work.

Because this method cannot correct all errors automatically, the potential exists for uncorrected errors to exist in the final string. Some of these errors could be fixed by combining this method with an autocomplete method similar to that proposed by Ryan et al. [17]. It is also possible that many errors do not need to be corrected in order for the user's intent to be conveyed as readers are able to understand text that contains errors. Depending on the application of the system, a certain error rate in the final output could be acceptable if

it is associated with an increased typing speed. The relationship between error rate in BCI communication output and reader understanding remains to be studied.

The amount of data acquired by the online pilot experiments was limited because of the number of configurations that needed to be tested. The results were therefore subject to significant variability as single errors could result in large changes in bit rate. The optimal threshold values also could not be computed for the online system as only two candidate values could be tested for each method. Finally, the difference in performance when subjects are required to correct errors was not explored. Additional online tests, possibly including multiple sessions with the same user, could provide more optimal configurations of the system and a better evaluation of online performance.

V. Conclusion

Typing with a P300 system can be modeled as a Markov process that can be indirectly observed through EEG response signals. The Viterbi algorithm effectively incorporates domain information into signal classification, which greatly improves the user's ability to create language. This study shows that incorporating this natural language information significantly improves the performance of a BCI communication system.

Acknowledgments

This work was supported by the NLM Training Grant T15-LM007356 (WS), the National Institute of Biomedical Imaging and Bioengineering Award Number K23EB014326 (NP), the UCLA Scholars in Translational Medicine Program (NP), and the UCLA Department of Neurosurgery Visionary Ball Fund.

References

1. Wolpaw J, Birbaumer N, McFarland D, Pfurtscheller G, Vaughan T. Brain-computer interfaces for communication and control. *Clin Neurophysiol.* 2002; 133:767–791. [PubMed: 12048038]
2. Farwell L, Donchin E. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalogr Clin Neurophysiol.* 1988; 70:510–512. [PubMed: 2461285]
3. Sellers E, Krusienski D, McFarland D, Vaughan T, Wolpaw J. A P300 event-related potential brain-computer interface (BCI): the effects of matrix size and inter stimulus interval on performance. *Biol Psychol.* 2006; 73:242–252. [PubMed: 16860920]
4. Townsend G, LaPallo B, Boulay C, Krusienski D, Frye G, Hauser C, Schwartz N, Vaughan T, Wolpaw J, Sellers E. A novel P300-based brain-computer interface stimulus presentation paradigm: moving beyond rows and columns. *Clin Neurophysiol.* 2010; 121:1109–1120. [PubMed: 20347387]
5. Jin J, Horki P, Brunner C, Wang X, Neuper C, Pfurtscheller G. A new P300 stimulus presentation pattern for EEG-based spelling systems. *Biomed Tech.* 2010; 55:203–210.
6. McFarland D, Sarnacki W, Townsend G, Vaughan T, Wolpaw J. The P300-based brain-computer interface (BCI): effects of stimulus rate. *Clin Neurophysiol.* 2011; 122:731–737. [PubMed: 21067970]
7. Lu J, Speier W, Hu X, Pouratian N. The effects of stimulus timing features on P300 speller performance. *Clin Neurophysiol vol.* 2012; 124(2):306–314.
8. Kaper M, Meinicke P, Grossekhoefer U, Lingner T, Ritter H. BCI competition 2003 - data set IIB: support vector machines for the P300 speller paradigm. *IEEE Trans Biomed Eng.* 2004; 50:1073–1076. [PubMed: 15188881]

9. Xu N, Gao X, Hong B, Miao X, Gao S, Yang F. BCI competition 2003 - data set Iib: enhancing P300 wave detection using ICA-based subspace projections for BCI Applications. *IEEE Trans Biomed Eng.* 2004; 51:1067–1072. [PubMed: 15188880]
10. Serby H, Yom-Tov E, Inbar G. An improved P300-based brain-computer interface. *IEEE Trans Neural Syst Rehabil Eng.* 2005; 13:89–98. [PubMed: 15813410]
11. Krusienski D, Sellers E, Cabestaing F, Bayouduh S, McFarland D, Vaughan T, Wolpaw J. A comparison of classification techniques for the P300 Speller. *J Neural Eng.* 2006; 3:299–305. [PubMed: 17124334]
12. Speier W, Arnold C, Lu J, Taira RK, Pouratian N. Natural language processing with dynamic classification improve P300 speller accuracy and bit rate. *J Neural Eng.* 2012; 9(1):016004. [PubMed: 22156110]
13. Park J, Kim K. A POMDP Approach to Optimizing P300 Speller BCI Paradigm. *IEEE Trans Neural Syst Rehabil Eng.* 2012; 20:584–594. [PubMed: 22510955]
14. Kindermans P, Verschore H, Verstraeten D, Schrauwen B. A P300 BCI for the Masses: Prior Information Enables Instant Unsupervised Spelling. *NIPS.* 2012; 25:719–727.
15. Hart-Davis, G. *Office 2010 Made Simple.* New York: Springer; 2011. Entering and Editing Text in Your Documents; p. 165-197.
16. Dunlop M, Crossan A. Predictive text entry methods for mobile phones. *Personal Technologies.* 2000:134–143.
17. Ryan D, Frye G, Townsend G, Berry D, Mesa SG, Gates N, Sellers E. Predictive spelling with a P300-based brain-computer interface: increasing the rate of communication. *Int J Hum-Comput Interact.* 2011; 27:69–84. [PubMed: 21278858]
18. Draper, N.; Smith, H. *Applied Regression Analysis.* 2. New York: Wiley; 1981.
19. Duda, R.; Hart, P.; Stork, D. *Pattern Classification.* 2. New York: Wiley; 2001.
20. Francis W, Kucera H. *Brown Corpus Manual.* 1979
21. Manning, C.; Schütze, H. *Foundations of Statistical Natural Language Processing.* Cambridge, MA: MIT Press; 1999.
22. Pierce, J. *An Introduction to Information Theory.* New York: Dover; 1980.

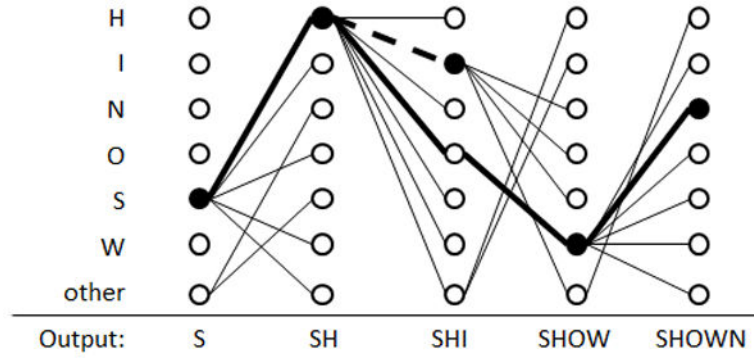


Fig. 1. Simplified Viterbi trellis for subject B spelling the word “shown.” At time $t = 3$, the character ‘I’ has the highest probability, resulting in the output “shi” after following the back pointers (dotted lines). At time $t = 4$, the character ‘W’ has the highest probability and the back pointers (bold lines) produce the output “show,” correcting the previous mistake.

Table 1

Target words for BCI trials. Each subject was shown a subset of between eight and 10 of these words.

AFTER	BLOCK	CLEAR	DAILY	FIRST
GIANT	HOURS	MINUS	NOTED	PANEL
SCORE	SHOWN	UNITS		

Table 2

Optimal selection rates, accuracies, and information transfer rates for the 15 subjects after optimizing on ITR in offline analysis.

Subject	SR (selections/minute)					ACC (%)					ITR (bits/minute)				
	SWLDA	NB	HMM	SWLDA	NB	SWLDA	NB	HMM	SWLDA	NB	HMM	SWLDA	NB	HMM	
A	8.07	8.33	9.80	93.33	100.00	95.56	36.13	43.08	45.87						
B	5.33	5.81	7.42	88.89	95.56	88.89	21.82	27.17	30.38						
C	3.96	4.09	8.45	80.00	95.00	67.50	13.54	18.92	21.91						
D	9.96	10.91	10.83	88.89	93.33	97.78	40.82	48.81	53.08						
E	3.98	5.83	5.20	95.56	84.44	97.78	18.64	21.83	25.47						
F	7.21	8.07	8.99	95.56	97.78	95.56	33.76	39.57	42.09						
G	5.33	6.03	6.56	94.00	96.00	92.00	24.18	28.48	28.60						
H	8.66	9.06	10.17	88.00	90.00	90.00	34.86	37.96	42.59						
I	5.08	6.36	9.38	78.00	86.00	78.00	16.68	24.58	30.77						
J	4.78	6.32	5.86	90.00	84.00	94.00	20.03	23.46	26.57						
K	3.98	8.24	6.68	84.00	68.00	80.00	14.80	21.63	22.85						
L	5.62	8.08	7.36	92.00	86.00	90.00	24.47	31.23	30.80						
M	4.13	6.61	6.19	82.00	82.00	84.00	14.72	23.57	22.98						
N	7.60	8.01	8.98	94.00	96.00	90.00	34.48	37.83	37.62						
O	4.40	7.53	6.28	88.00	78.00	84.00	17.71	24.70	23.33						
average	5.87	7.28	7.88	88.82	88.81	88.34	24.44	30.19	32.33						

Table 3

Optimal selection rates, accuracies, and information transfer rates for the five subjects in online trials.

Subject	SR (selections/minute)				ACC (%)				ITR (bits/minute)					
	SWLDA	NB	HMM	SWLDA	SWLDA	NB	HMM	SWLDA	SWLDA	NB	HMM	SWLDA	NB	HMM
A	5.52	9.02	10.00	89.29	83.33	83.33	95.24	22.79	33.05	46.51				
B	5.52	7.07	8.11	75.00	82.86	82.86	90.24	16.98	25.65	34.12				
C	3.28	8.10	6.96	94.12	85.37	85.37	88.57	14.91	30.93	28.35				
D	5.52	11.09	10.90	100.00	83.64	83.64	94.44	28.53	40.90	49.86				
E	5.52	11.50	11.80	100.00	78.95	78.95	93.22	28.53	38.49	52.69				
average	5.07	9.35	9.55	91.68	82.83	82.83	92.34	22.35	33.80	42.31				

Table 4

Example online output for each of the tested methods. Each is the result of subject D attempting to spell “Heroes in a half shell turtle power.” HMM* would be the output of the HMM method if the errors were not corrected.

Method	Output
SWLDA	HEROES IN A HALF SHELL TURTLE
NB	HEROE RIN A HARE SHELL TURTLE PREER HERIES IN AINALF SH
HMM *	TEROESINAHAL6HERLTURTLEPOWEDHEROESINAHALFLG
HMM	HEROES IN A HALL THELL TURTLE POWED HEROES IN A HALF S