Moving away from error-related potentials to achieve spelling correction in P300 spellers

Boyla O. Mainsah, Kenneth D. Morton, *Member, IEEE,* Leslie M. Collins, *Senior Member, IEEE,* Eric W. Sellers, and Chandra S. Throckmorton

***Abstract*—P300 spellers can provide a means of communication for individuals with severe neuromuscular limitations. However, its use as an effective communication tool is reliant on high P300 classification accuracies (>70%) to account for error revisions. Error-related potentials (ErrP), which are changes in EEG potentials when a person is aware of or perceives erroneous behaviour or feedback, have been proposed as inputs to drive corrective mechanisms that veto erroneous actions by BCI systems. The goal of this study is to demonstrate that training an additional ErrP classifier for a P300 speller is not necessary, as we hypothesize that error information is encoded in the P300 classifier responses used for character selection. We perform off-line simulations of P300 spelling to compare ErrP and non-ErrP based corrective algorithms. A simple dictionary correction based on string matching and word frequency significantly improved accuracy (35-185%), in contrast to an ErrP-based method that flagged, deleted and replaced erroneous characters (-50-0%). Providing additional information about the likelihood of characters to a dictionary-based correction further improves accuracy. Our Bayesian dictionary-based correction algorithm that utilizes P300 classifier confidences performed comparably (44-416%) to an oracle ErrP dictionary-based method that assumed perfect ErrP classification (43-433%).**

***Index Terms*—Electroencephalogram, Brain-Computer Interface, P300 Speller, Error-Related Potential, Noisy Channel Model.**

I. Introduction

THE P300 speller is a brain-computer interface (BCI) that exploits event-related potentials (ERP) in electroencephalography (EEG) data to enable users to control a word processing program [1]. It has been recommended that P300 classification rates perform with accuracies greater than 70% for effective communication [2], as spelling correction requires at least two selective actions: correctly selecting *backspace* and reselecting the intended character. Alternatively, system usability can be improved if erroneously spelled characters can be automatically detected and deleted without further user action, saving the time needed to select a *backspace* command.

One method that has been proposed for actively detecting errors is to detect error-related potentials. Error-related potentials (ErrP) are changes in the EEG potentials after a person becomes aware of or perceives erroneous behavior [3].

Manuscript received 27 January 2014; revised 5 June 2014 and 15 August 2014; accepted 10 November 2014.

This work is supported in part by NIH/NIDCD grant number R33DC010470-03

B. O. Mainsah, K. D. Morton Jr., L. M. Collins, and C. S. Throckmorton are with the Department of Electrical and Computer Engineering at Duke University, Durham, NC 27708, USA (Author contact email: leslie.collins@duke.edu; phone: 919 660 5260).

E. W. Sellers is with the Department of Psychology at East Tennessee State University, Johnson City, TN 37614, USA.

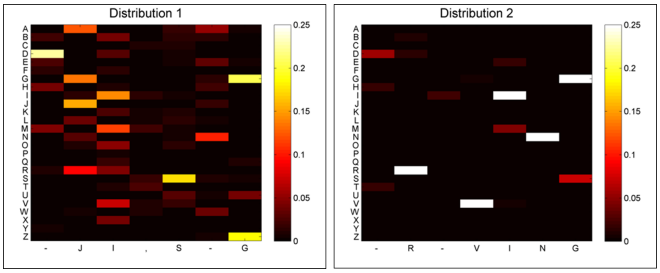

Fig. 1. Distribution of character probabilities post-data collection for the word **DRIVING**, simulated from EEG data from two P300 speller sessions. The x-axis labels show the characters selected by the P300 speller, yielding words, **-JI,S-G** (left) and **-R-VING** (right), with the corresponding probabilities of alphabet characters (columns). Ideally, the character with the highest probability should correspond to the target, e.g. most characters in distribution 2. For erroneous characters in both distributions, target characters are usually among those with the next highest probabilities. Probability values are clipped for visualization purposes and non-alphabetic grid characters are not displayed.

ErrP detection has been suggested as input to drive corrective mechanisms that veto erroneous actions by BCIs. However, the limited on-line studies with ErrP-driven corrective mechanisms in P300 spellers have produced mixed results [4]–[7].

A common complaint with training ErrP classifiers for P300 spellers is the long time required to obtain enough ErrP classifier training data. For example, using a paradigm such as that proposed by Townsend *et al.* [8] that presents 24 flashes/sequence and a typical amount of data collection of 5 sequences/character would result in 120 labeled samples with which to train the P300 classifier for a single spelled character. On the other hand, presenting that spelled character only yields 1 labeled sample for ErrP classifier training. The lack of adequate training data can negatively affect the potential benefit of using ErrP detection for automatic error deletion since efficacy depends on the accuracy of detection. Further, single trial ERP detection (i.e. the user's response to a single erroneous character) within noisy EEG data can be challenging due to the low signal-to-noise ratio of ERPs. Average on-line detection performance of ErrP classifiers for automatic character deletion in P300 spellers has ranged from 60-90% accuracy, with 40-60% sensitivity (hit rate) and 80-90% specificity [4]–[7].

In this study, we consider whether an alternative approach that does not rely on detecting ErrPs has the potential to provide similar or better error correction. The positional context of erroneous characters can be used to infer the user's intended word from a dictionary of words via string matching [9]. String matching searches for the words that match the misspelled word within a certain number of edits (termed edit distance),

e.g. tar would match txr with one edit, the substitution of x with a. In addition, the cumulative P300 classifier outputs prior to character selection contain some information about the likelihood of possible letters being the target at each position in the word [8], [10]. This uneven distribution of character classifier outputs is useful when more than one word match is obtained from a dictionary after string matching e.g. [11].

Fig. 1 shows two example probability distributions of alphabet characters post-data collection for the word **DRIVING**, obtained by using EEG data from P300 speller sessions of two different participants to simulate spelling, yielding the words: **-JI,S-G** and **-R-VING**. The 9×8 Townsend *et al.* P300 speller grid [8] used in this study consists of alphanumeric characters and 36 additional command/grammar options e.g."Del," "Home." The command options were disabled for the spelling sessions, and if selected, were represented by a hyphen. Attempting to correct misspellings from string matching alone would be difficult, since there are several matches that can be obtained within the same edit distance e.g. CRAVING, DRIVING, PROVING etc., for **-R-VING**. As the number of errors increases, e.g. **-JI,S-G**, the number of possible matches can increase. Therefore, a method of choosing from alternative words is required. A common method is word frequency. In this study, we also consider using the probability of each character being the target character. As can be observed in Fig. 1, while the target character may not have the highest probability, often one of the next most probable characters is the target. Thus, the character probabilities can be used to weight characters in word choices when performing spelling correction, e.g. in the first position of both distributions, **D** has the highest alphabet probability.

We hypothesize that training an additional ErrP classifier to flag erroneous characters is not needed, since as shown above, some error information is encoded in the cumulative P300 classifier responses. In this study, we compare the performance of spelling correction with ErrP and non-ErrP based corrective algorithms. We perform offline analyses to compare the improvement in accuracy from the raw P300 speller character selections using various corrective algorithms.

II. Methods

A. EEG Dataset

The dataset was obtained at East Tennessee State University for a study approved by the university's Institutional Review Board. Participants were numbered in the order they were recruited (n = 19). The open source BCI2000 software package was used for stimulus presentation and data collection [12]. The checkerboard paradigm was used on a 9×8 grid [8]. EEG responses were measured using a 32-channel electrode cap, with the left and right mastoids used for ground and reference electrodes, respectively. The EEG signals were amplified, digitized at 256 Hz, and filtered between 0.5 - 30 Hz.

Participants underwent two P300 speller sessions: a first session to collect data to train a P300 classifier and a second to collect data to train an ErrP classifier. During the first session, participants spelled four 5-letter words with five sequences/character (2 target flashes out of 24 flashes/sequence). During the second session, the trained P300 classifier was not used online. Participants spelled 15 phrases of 20 characters, each with fake feedback presented at an error rate of 20%. To speed up data collection for the ErrP classifier, only one sequence/character was used prior to presenting the fake feedback. Off-line signal analysis and spelling correction were performed using MATLAB software (The MathWorks, Inc.).

B. Signal Analysis and Classification

1) P300 Classification: Using the EEG data from the first session, features were extracted to train a stepwise linear discriminant analysis (SWLDA) classifier [13]. The likelihood probability density functions (pdf), $p(x|H_0)$ and $p(x|H_1)$, of target and non-target scores, respectively, were generated by using kernel density estimation to smooth out the histogram of the grouped scores, and the $p(x|H_0)$ and $p(x|H_1)$ pdfs were used in the Bayesian spelling correction algorithm.

2) ErrP Classification: Using the EEG data post-character feedback from the second training session, features were extracted to train a linear discriminant analysis (LDA) classifier, with shrinkage [14], with leave-one-word-out cross-validation. The likelihood pdfs, $p(s|H_c)$ and $p(s|H_e)$, of correct and erroneous character scores, respectively, were generated by using kernel density estimation to smooth out the histograms of the grouped scores. The $p(s|H_c)$ and $p(s|H_e)$ pdfs were used in the ErrP classifier spelling correction algorithms.

C. P300 Spelling and ErrP Classifier Simulation

The P300 classifier trained from the first spelling session data was applied to the EEG data of the second session to simulate P300 spelling. In the example in Fig. 2, a user intends to spell the word $C = (c_1, c_2, ..., c_T)$. For a spelled word, $W = (w_1, w_2, ..., w_T)$, the selected character, w_t, was the character with the maximum cumulative P300 classifier score. The P300 classifier also outputs a $\boldsymbol{Q}^{N\times T}$ matrix, which can be the cumulative classifier score rankings or probabilities of grid characters prior to character selection. Each column in the $\boldsymbol{Q}$ matrix, $\boldsymbol{Q}_t$, corresponds to labeled entries of the N characters in the grid for the t^{th} spelled character. Examples of $\boldsymbol{Q}$ matrices are shown in Fig. 1.

If applicable, the trained ErrP classifier was applied to features extracted from a time window of EEG data after the feedback was presented to the user. The ErrP classifier returns a score vector, $\boldsymbol{S} = [s_1, s_2, ..., s_T]$, which was used to calculate the ErrP classifier confidences, $\boldsymbol{\Pi} = [\pi_1, \pi_2, ..., \pi_T]$:

$$\pi_t = \frac{p(s_t|H_c)A_{pr}}{p(s_t|H_c)A_{pr} + p(s_t|H_e)(1 - A_{pr})} \tag{1}$$

where π_t is the confidence that the selected character, w_t, is correct; $p(s_t|H_c)$ and $p(s_t|H_e)$ are the likelihoods that the ErrP classifier score, s_t, is generated from a correct character and incorrect character (hence ErrP elicited), respectively; A_{pr} is the projected accuracy calculated from the P300 training data of the first session, according to Colwell *et al.* [15].

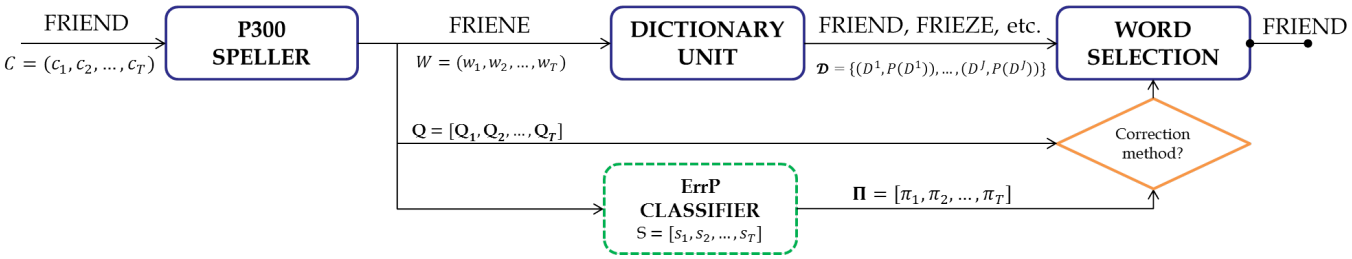

Fig. 2. Flowchart for proposed spelling correction in the P300 speller. A user intends to spell the word C. Using a trained P300 classifier and EEG data, the P300 speller outputs the spelled word, W, and a matrix of character P300 classifier score rankings or probabilities, $\boldsymbol{Q}^{N\times T}$, ($N$ = number of grid characters, T = length of the spelled word). In the dictionary unit, a list of probable words, $\boldsymbol{\mathcal{D}}$, based on a string metric function is generated from a vocabulary, with the corresponding prior probabilities, $\left(D^j, P(D^j)\right)$, obtained from a text corpus. If an ErrP classifier is used, after character selection feedback is presented to the user, the ErrP classifier computes classifier scores, $\boldsymbol{S}^{1\times T}$, which are used to calculate the ErrP classifier confidences, $\boldsymbol{\Pi}^{1\times T}$. Using the word prior probabilities, $P(D^j)$, the $\boldsymbol{Q}$ matrix or $\boldsymbol{\Pi}$ vector, the user's intended word choice, $\hat{C}$, is estimated.

D. Spelling Correction

For the spelled word, W, a set of possible word choices and their corresponding unigram probabilities, $\boldsymbol{\mathcal{D}} = \{(D^1, P(D^1)), (D^2, P(D^2)), ..., (D^J, P(D^J))\}$, was generated from a dictionary and used to estimate the word the user intended to spell, C (see Fig. 2). The dictionary vocabulary ($\approx 30,000$ words) was created from a modified corpus compiled by Norvig [16] and the frequency count of words were smoothed to obtain word unigram probabilities. The word choices were limited to words of the same length with minimum Levenshtein edit distance from the spelled word. Their unigram probabilities, $P(D^j)$, provided an estimate of the prior probability of being the user's intended word. The Levenshtein distance is the minimum number of single-character edits, insertions, deletions and substitutions, needed to convert one string to another [17].

For some algorithms, a noisy channel model was used for spelling correction [18], [19]:

$$P(D^j|W, \boldsymbol{Q}/\boldsymbol{\Pi}) \propto P(W|D^j, \boldsymbol{Q}/\boldsymbol{\Pi})P(D^j) \tag{2}$$

$$\hat{C} = \arg\max_{D^j \in \boldsymbol{\mathcal{D}}} P(W|D^j, \boldsymbol{Q}/\boldsymbol{\Pi})P(D^j) \tag{3}$$

where $P(D^j|W, \boldsymbol{Q}/\boldsymbol{\Pi})$ is the posterior probability of the word choice, D^j, given the spelled word, W and $\boldsymbol{Q}$ matrix from the P300 classifier or $\boldsymbol{\Pi}$ vector from the ErrP classifier; and $P(W|D^j, \boldsymbol{Q}/\boldsymbol{\Pi})$ is the likelihood of the spelled word given the word choice, D^j, and $\boldsymbol{Q}/\boldsymbol{\Pi}$.

Three non ErrP-based detection methods were compared to three ErrP-based detection methods. The non-ErrP detection based methods consisted of a dictionary look-up with different word selection methods: word frequency; a method proposed by Ahi *et al.* for word ranking [11]; and a Bayesian method based on the P300 classifier character probabilities (methods 1-3). The ErrP detection based methods include: an oracle method in which perfect ErrP detection was assumed; a method proposed by Margaux *et al.* for using ErrP detection for error correction [6]; and a method for which the ErrP detection classifier confidences are used in the noisy channel model (methods 4-6).

1) Simple dictionary: The word with the highest unigram probability in $\boldsymbol{\mathcal{D}}$, was selected as the target word estimate:

$$\hat{C} = \arg\max_{D^j \in \boldsymbol{\mathcal{D}}} P(D^j) \tag{4}$$

2) Ahi et al., 2011: The classifier scores were used to rank each word choice to obtain an estimate of the user's intended word [11]. The word with the minimum cost was selected as the target word estimate:

$$r(D^j) = \sum_{t=1}^{T} q_t^{l(D_t^j)} \tag{5}$$

$$\hat{C} = \arg\min_{D^j \in \boldsymbol{\mathcal{D}}} r(D^j) \tag{6}$$

where D_t^j is the t^{th} letter of the word D^j, $l(D_t^j)$ is the grid label for D_t^j and $q_t^l(D_t^j)$ is the rank of the classifier score of D_t^j, obtained from the t^{th} column of the $\boldsymbol{Q}$ matrix, $\boldsymbol{Q}_t$. The $\boldsymbol{Q}$ matrix for this algorithm consists of the cumulative P300 classifier score rankings of characters in the spelled word.

3) Bayesian: The cumulative character scores were not used for character selection. Instead, each character was assigned a uniform prior probability of being the target and a Bayesian approach was used to update the character probabilities with each EEG flash data [20]. The character with the maximum probability at the end of the Bayesian updates was selected as the user's intended choice, w_t. The column entries in the $\boldsymbol{Q}$ matrix thus consisted of the final Bayesian character probabilities. The noisy channel model (2, 3) was used for spelling correction to estimate the user's intended word:

$$P(W|D^j, \boldsymbol{Q}) = \left(\prod_{t=1}^{T} q_t^{l(D_t^j)}\right) P(D^j) \tag{7}$$

where $q_t^l(D_t^j)$ is the Bayesian character probability of D_t^j, obtained from the t^{th} column of the $\boldsymbol{Q}$ matrix, $\boldsymbol{Q}_t$.

4) Oracle ErrP Classifier: The oracle ErrP classifier was used to infer the upper bound on the performance of spelling correction with perfect ErrP classification, i.e. returns "0" for correctly spelled characters and "1" for erroneous characters. The set of words in $\boldsymbol{\mathcal{D}}$ was narrowed to words with substitutions only at erroneous character locations. The word with the highest unigram probability in $\boldsymbol{\mathcal{D}}$ was selected as the target word estimate, according to (4).

5) Margaux et al., 2012: The ErrP classifier confidences, $\boldsymbol{\Pi}$, were compared against the projected accuracy, A_{pr} [15], calculated from the EEG data of the first P300 speller session. If an ErrP classifier confidence, π_t, was less than the projected

accuracy, A_{pr}, character w_t was substituted with the character that had the 2^{nd} P300 classifier score rank in $\boldsymbol{Q}_t$ [6].

6) ErrP Classifier: The noisy channel model (2, 3) was used for spelling correction to estimate the user's intended word and was based on the ErrP confidences:

$$P(D^j|W,\mathbf{\Pi}) = \left[\prod_{t=1}^{T} \pi_t^{\left(\delta_{w_t,D_t^j}\right)} \left(\frac{1-\pi_t}{N-1}\right)^{\left(1-\delta_{w_t,D_t^j}\right)}\right] P(D^j) \tag{8}$$

where δ_{w_t,D_t^j} is the Kronecker delta, where $\delta = 1$ when $w_t = D_t^j$, and $\delta = 0$ when $w_t \neq D_t^j$; π_t is the ErrP classifier confidence; and $\frac{1-\pi_t}{N-1}$ is the remaining ErrP classifier confidence that is evenly distributed across the remaining $N-1$ characters in the grid.

E. Performance measures

The character and word accuracies for the P300 speller simulation, with and without the spelling correcton algorithms were calculated for each participant. Statistical significance was tested using a repeated measures ANOVA.

III. Results

The character and word accuracies for the raw P300 speller and with spelling correction were calculated. Fig. 3(A) and (B) shows pooled participant results. Statistical analyses for character and word accuracy revealed a significant difference in the means of at least two algorithms ($p < 0.05$), and pairwise comparisons are shown in Table I and II. The performance percentage improvements reported are with respect to the raw P300 speller character accuracy. Participant-specific results are shown in Fig. 3(C), ordered according to raw P300 speller character accuracy (also see Table III and IV).

Fig. 3(A) compares character-based correction with the ErrP classifier to word-based correction with a simple dictionary correction. It can be observed that correcting whole words with errors is more beneficial as it utilizes the positional context of errors to generate word alternatives. Even with just one sequence of data prior to character selection, a simple dictionary correction was able to yield a significant increase in participant accuracy, with 35-185% improvement in character accuracy.

Successfully deleting and replacing erroneous characters that are flagged by an ErrP classifier, as in Margaux *et al.*, requires high discriminability by the ErrP classifier. At these ErrP detection performances, utilizing the Margaux *et al.* method negatively impacted participant accuracy, ranging from -47 to 0% decrease in character accuracy. It is possible that the Margaux *et al.* correction method was adversely affected by the limited amount of data collection prior to character selection, as more data could have led to sparser character distributions where likely and unlikely characters are better separated. However, there is no guarantee that if the ErrP classifier correctly flags and deletes an erroneous character, substitution with the next most probable character will correspond with the target. Nonetheless, these results highlight the benefit of including language information, especially under the challenging condition of limited training data.

Fig. 3(B) shows the potential benefit of adding information about the confidence in each character to the language-based error correction of the simple dictionary search. The ErrP classifier correction method (35-190%) is comparable to a simple dictionary correction, suggesting no additional benefit in attempting to correct errors at these ErrP detection accuracies. However, with perfect ErrP detection, a significant benefit occurs (43-433%), suggesting that the knowledge of incorrect characters can be beneficial to word correction. Relying on cumulative P300 classifier score rankings of letters to rank words, as in Ahi *et al.*, has some benefit to word correction (37-416%); however, the Bayesian approach that uses a noisy channel model further improves performance (44-416%). Furthermore, performance with the noisy channel model with Bayesian character probabilities is similar to that with perfect ErrP detection. This suggests that training an additional ErrP classifier to flag erroneous characters in the P300 speller may not be necessary as spelling correction can be achieved with a dictionary by utilizing the error information that is encoded in the P300 classifier responses.

IV. Discussion

ErrP detection requires the collection of a substantial amount of training data in order to be accurate, and the accuracy of the detection drives the efficacy of corrective mechanisms based on ErrPs. The difference between the accuracies of the trained ErrP-based correction method and the oracle ErrP-based correction method was statiscally significant, both for characters (difference of approximately 25 characters correct) and words (difference of approximately 4 words correct). This suggests that additional training data would be required to achieve the full potential of the ErrP detection-based correction method. However, by relying on language information and BCI outputs, equivalent performance to the oracle ErrP-based correction method was achieved without the requirement for additional training data. Thus, the Bayesian correction method has the potential to improve accuracy at a much reduced cost in time and effort.

The Bayesian correction algorithm has the further advantage of being applicable to other ERP-based spelling BCIs with probabilistic data collection algorithms and it can be incorporated within any probabilistic-based spelling correction algorithm. Spell-checking and correction algorithms have been widely studied for other applications and can be exploited for BCI spelling applications [21]. For example, while we used unigram word probabilities, additional context within sentences can be provided via higher order n-gram language models for context-based spelling correction, especially for detecting and correcting real-word errors.

While an on-line implementation was beyond the scope of this study, the oracle ErrP-based correction algorithm provided an estimation of the upper bound on an ErrP-based system. In this offline analysis, the Bayesian correction method achieved similar performance to the upper bound, suggesting the potential for correction without ErrP detection. However,

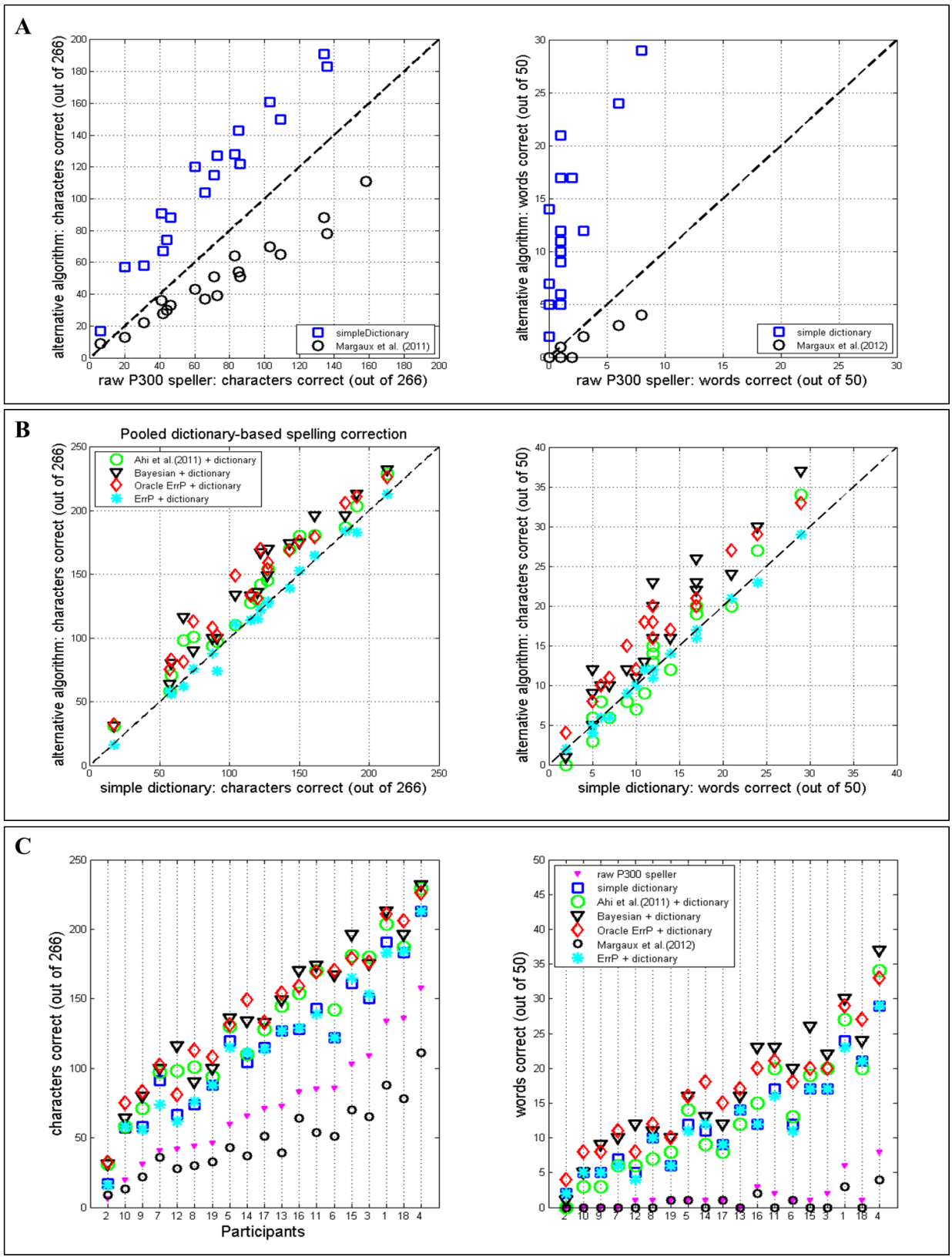

Fig. 3. Character and word accuracies for the raw P300 speller outputs and spelling correction algorithms. Fig. 3(A) shows pooled participant results comparing character-based (Margaux *et al.*) and word-based (simple dictionary) spelling correction. It can be observed that using the positional context of errors via a priori knowledge of the user's language in word-based spelling correction noticeably improves P300 speller word and character accuracy. Fig. 3(B) shows pooled participant results comparing the effect of including additional information from either the ErrP or P300 classifier to a simple dictionary spelling correction, as characters in word choices are differently weighted. Fig. 3(C) shows participant-specific results, ordered by increasing raw P300 speller character accuracy.

TABLE I
STATISTICAL COMPARISON BETWEEN CORRECTION ALGORITHMS: CHARACTER PERFORMANCE

ALGORITHM Mean ± Std	Raw P300 speller 73.37 ± 41.07	Simple dictionary 116.26 ± 50.61	Ahi *et al.* + dictionary 132.10 ± 52.29	Bayesian + dictionary 139.79 ± 53.64	Oracle ErrP + dictionary 139.84 ± 51.60	Margaux *et al.* 48.53 ± 26.13	ErrP + dictionary 115.00 ± 51.20
Raw P300 speller		↓	↓	↓	↓	↑	↓
Simple dictionary			↓	↓	↓	↑	
Ahi *et al.* + dictionary						↑	↑
Bayesian + dictionary						↑	↑
Oracle ErrP + dictionary						↑	↑
Margaux *et al.*							↓
ErrP + dictionary							

Character accuracy is out of 266 characters. Analysis performed using repeated measures ANOVA (p-value < 0.05), with Bonferroni adjustment for pair-wise comparisons.
LEGEND: ↑, significantly higher; ↓, significantly lower. Legend entries are interpreted row-wise.
Example: Entry ↑ in (x, y) means performance of the algorithm in row x is significantly higher than that in column y.

TABLE II
STATISTICAL COMPARISON BETWEEN CORRECTION ALGORITHMS: WORD PERFORMANCE

ALGORITHM Mean ± Std	Raw P300 speller 1.58 ± 2.09	Simple dictionary 12.36 ± 7.12	Ahi *et al.* + dictionary 12.84 ± 8.80	Bayesian + dictionary 16.84 ± 9.04	Oracle ErrP + dictionary 16.58 ± 7.70	Margaux *et al.* 0.68 ± 1.15	ErrP + dictionary 12.10 ± 7.11
Raw P300 speller		↓	↓	↓	↓		↓
Simple dictionary				↓	↓	↑	
Ahi *et al.* + dictionary				↓		↑	
Bayesian + dictionary						↑	↑
Oracle ErrP + dictionary						↑	↑
Margaux *et al.*							↓
ErrP + dictionary							

Word accuracy is out of 50 words. Analysis performed using repeated measures ANOVA (p-value < 0.05), with Bonferroni adjustment for pair-wise comparisons.
LEGEND: ↑, significantly higher; ↓, significantly lower. Legend entries are interpreted row-wise.
Example: Entry ↑ in (x, y) means performance of the algorithm in row x is significantly higher than that in column y.

the Bayesian spelling algorithm requires further development prior to on-line BCI spelling applications. The performance of dictionary-based correction is dependent on the language models developed from a compiled corpus. A user-specific body of text can provide more language context and it can be updated and smoothed periodically to handle out-of-vocabulary words. Another issue is the detection of word boundaries/white space prior to performing spelling correction. Most P300 speller studies design their spelling tasks with single words and in this study, we extracted words from phrases, hence the target word length is known *a priori*. In addition, dictionary-based spelling correction is not applicable to numbers or command options in the speller grid. Natural language processing tools like word segmentation/tokenization [22] or techniques from optical character recognition [23] can be exploited to further improve the performance of BCI spellers for more practical use for the target BCI population.

V. CONCLUSION

This study demonstrates that spelling correction can be achieved in BCI spellers without the large costs in data and time associated with ErrP-driven corrective mechanisms. Instead, a new spelling correction algorithm is developed, the noisy channel model with Bayesian character probabilities, which combines probabilistic P300 classifier information and dictionary-based suggestions to achieve a significant increase in character/word accuracy (44-416%) from the raw P300 speller outputs. This algorithm achieves comparable performance to an ErrP-based correction method for which perfect ErrP detection is assumed (43-433%), suggesting that the Bayesian method may provide a more reliable approach to spelling correction than developing an ErrP-based classifier.

ACKNOWLEDGMENT

We would like to thank the participants who dedicated their time for data collection. The authors would also like to thank the two anonymous reviewers for their respective comments.

REFERENCES

[1] L. A. Farwell and E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalography and Clinical Neurophysiology*, vol. 70, no. 6, pp. 510–23, 1988.
[2] F. Nijboer *et al.*, "A p300-based brain-computer interface for people with amyotrophic lateral sclerosis," *Clinical Neurophysiology*, vol. 119, no. 8, pp. 1909–1916, 2008.
[3] M. Falkenstein, J. Hohnsbein, J. Hoormann, and L. Blanke, "Effects of crossmodal divided attention on late erp components .2. error processing in choice reaction tasks," *Electroencephalography and Clinical Neurophysiology*, vol. 78, no. 6, pp. 447–455, 1991.
[4] B. Dal Seno, M. Matteucci, and L. Mainardi, "Online detection of p300 and error potentials in a bci speller," *Comput Intell Neurosci*, 2010.
[5] N. M. Schmidt, B. Blankertz, and M. S. Treder, "Online detection of error-related potentials boosts the performance of mental typewriters," *BMC Neuroscience*, vol. 13, p. 19, 2012.
[6] P. Margaux, M. Emmanuel, D. Sebastien, B. Olivier, and M. Jeremie, "Objective and subjective evaluation of online error correction during p300-based spelling," *Advances in Human-Computer Interaction*, vol. 2012, p. 13, 2012.
[7] M. Spüler, M. Bensch, S. Kleih, W. Rosenstiel, M. Bogdan, and A. Kübler, "Online use of error-related potentials in healthy users and people with severe motor impairment increases performance of a p300-bci," *Clinical Neurophysiology*, vol. 123, no. 7, pp. 1328–1337, 2012.
[8] G. Townsend *et al.*, "A novel p300-based brain-computer interface stimulus presentation paradigm: moving beyond rows and columns," *Clinical Neurophysiology*, vol. 121, no. 7, pp. 1109–20, 2010.

TABLE III
PARTICIPANT ALGORITHM PERFORMANCE: CHARACTER ACCURACY (OUT OF 266)

PARTICIPANT	Raw P300 speller	Simple dictionary	Ahi *et al.* + dictionary	Bayesian + dictionary	Oracle ErrP + dictionary	Margaux *et al.*	ErrP + dictionary
2	6	17	31	31	32	9	16
10	20	57	58	64	75	13	58
9	31	58	71	80	83	22	56
7	41	91	97	100	102	36	74
12	42	67	98	116	81	28	62
8	44	74	101	90	113	30	76
19	46	88	94	100	108	33	88
5	60	120	130	136	131	43	115
14	66	104	110	134	149	37	111
17	71	115	128	133	133	51	114
13	73	127	145	149	154	39	127
16	83	128	154	170	159	64	129
11	85	143	170	174	169	54	139
6	86	122	142	167	170	51	122
15	103	161	181	196	179	70	165
3	109	150	180	175	176	65	153
1	134	191	204	213	211	88	183
18	136	183	187	196	206	78	184
4	158	213	229	232	226	111	213

TABLE IV
PARTICIPANT ALGORITHM PERFORMANCE: WORD ACCURACY (OUT OF 50)

PARTICIPANT	Raw P300 speller	Simple dictionary	Ahi *et al.* + dictionary	Bayesian + dictionary	Oracle ErrP + dictionary	Margaux *et al.*	ErrP + dictionary
2	0	2	0	1	4	0	2
10	0	5	3	5	8	0	5
9	0	5	3	9	8	0	5
7	0	7	6	10	11	0	6
12	1	5	6	12	8	0	4
8	1	10	7	11	12	0	10
19	1	6	8	10	10	1	6
5	1	12	14	16	16	1	11
14	1	11	9	13	18	0	12
17	1	9	8	12	15	1	9
13	0	14	12	16	17	0	14
16	3	12	15	23	20	2	12
11	2	17	20	23	21	0	16
6	1	12	13	20	18	1	11
15	1	17	19	26	20	0	17
3	2	17	20	22	20	0	17
1	6	24	27	30	29	3	23
18	1	21	20	24	27	0	21
4	8	29	34	37	33	4	29

[9] D. Jurafsky and J. H. Martin, *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition*, 2nd ed., ser. Prentice Hall series in artificial intelligence. Upper Saddle River, N.J.: Pearson Prentice Hall, 2009.

[10] R. Fazel-Rezai, "Human error in p300 speller paradigm for brain-computer interface," *Conf Proc IEEE Eng Med Biol Soc*, vol. 2007, pp. 2516–9, 2007.

[11] S. T. Ahi, H. Kambara, and Y. Koike, "A dictionary-driven p300 speller with a modified interface," *IEEE Trans Neural Syst Rehabil Eng*, vol. 19, no. 1, pp. 6–14, 2011.

[12] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw, "Bci2000: Development of a general purpose brain-computer interface (bci) system," *Society for Neuroscience Abstracts*, vol. 27, no. 1, p. 168, 2001.

[13] D. J. Krusienski, E. W. Sellers, D. J. McFarland, T. M. Vaughan, and J. R. Wolpaw, "Toward enhanced p300 speller performance," *Journal of Neuroscience Methods*, vol. 167, no. 1, pp. 15–21, 2008.

[14] B. Blankertz, S. Lemm, M. Treder, S. Haufe, and K.-R. Müller, "Single-trial analysis and classification of erp components - a tutorial," *NeuroImage*, vol. 56, no. 2, pp. 814–825, 2011.

[15] K. Colwell, C. Throckmorton, L. Collins, and K. Morton, "Projected accuracy metric for the p300 speller," *IEEE Trans Neural Syst Rehabil Eng*, no. 99, 2014.

[16] P. Norvig, "How to write a spelling corrector," http://norvig.com/spell-correct.html, 2007.

[17] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Cybernetics and Control Theory*, vol. 10, no. 8, pp. 707–710, 1966.

[18] M. D. Kernighan, K. W. Church, and W. A. Gale, "A spelling correction program based on a noisy channel model," in *Proceedings of the 13th conference on Computational Linguistics*, vol. 2.

[19] E. Mays, F. J. Damerau, and R. L. Mercer, "Context based spelling correction," *Information Processing and Management*, vol. 27, no. 5, pp. 517 – 522, 1991.

[20] C. S. Throckmorton, K. A. Colwell, D. B. Ryan, E. W. Sellers, and L. M. Collins, "Bayesian approach to dynamically controlling data collection in p300 spellers," *IEEE Trans Neural Syst Rehabil Eng*, vol. 21, no. 3, pp. 508–17, 2013.

[21] K. Kukich, "Techniques for automatically correcting words in text," *ACM Computing Surveys*, vol. 24, no. 4, pp. 377–439, 1992.

[22] D. D. Palmer, *Tokenisation and sentence segmentation*. Marcel Dekker, Inc., New York, USA, 2000.

[23] S. Mori, H. Nishida, and H. Yamada, *Optical character recognition*. John Wiley & Sons, Inc., 1999.