Automatic Classification of Auto-correction Errors in Predictive Text Entry Based on EEG and Context Information

Felix Putze
University of Bremen
Bibliotheksstraße 1
Bremen 28359, Germany
felix.putze@uni-bremen.de

Maik Schünemann
University of Bremen
Bibliotheksstraße 1
Bremen 28359, Germany
maikschuenemann@gmail.com

Tanja Schultz
University of Bremen
Bibliotheksstraße 1
Bremen 28359, Germany
tanja.schultz@uni-bremen.de

Wolfgang Stuerzlinger
Simon Fraser University
250-13450 102nd Avenue
Surrey, BC V3T 0A3, Canada
w.s@sfu.ca

ABSTRACT

State-of-the-art auto-correction methods for predictive text entry systems work reasonably well, but can never be perfect due to the properties of human language. We present an approach for the automatic detection of erroneous auto-corrections based on brain activity and text-entry-based context features. We describe an experiment and a new system for the classification of human reactions to auto-correction errors. We show how auto-correction errors can be detected with an average accuracy of 85%.

CCS CONCEPTS

• **Human-centered computing** → **User models**; **User interface design**;

KEYWORDS

Predictive text entry; Brain-Computer Interface; EEG; Error Potentials; Context Information

ACM Reference format:
Felix Putze, Maik Schünemann, Tanja Schultz, and Wolfgang Stuerzlinger. 2017. Automatic Classification of Auto-correction Errors in Predictive Text Entry Based on EEG and Context Information. In *Proceedings of ICMI '17, Glasgow, United Kingdom, November 13–17, 2017,* 9 pages.
https://doi.org/10.1145/3136755.3136784

Today, predictive text entry systems are omnipresent on mobile phones and tablet computers. Those systems acknowledge that typing on a small keyboard with limited tactile feedback results in a higher error rate compared to typing on a physical computer keyboard. Thus, they provide auto-correction mechanisms to automatically replace mistyped words. Auto-correction methods have improved substantially over the last decade and nowadays can even use large-scale machine learning approaches for further improvements. However, even sophisticated state-of-the-art auto-correction approaches have error rates around 5% [11]. Given that we must expect a substantial fraction of typed words to contain at least one wrong letter[1], we must assume a high prevalence of auto-corrections – and therefore, auto-correction errors – during most text entry situations. Anecdotally, the impact of such errors is also documented by the existence of dedicated websites like http://damnyouautocorrect.com which show that auto-correction can still go wrong, generating unexpected results ranging from hilarious to disastrous. The development of methods for the mitigation of auto-correction errors by commercial enterprises [19] is another indicator for the necessity of handling such events in a less intrusive and cognitively exhausting way.

In this paper, we propose a novel technology to enhance auto-correction in a text entry system, which *actively* detects auto-correction errors as soon as they are perceived by the user. This is done by analyzing the user's cognitive processing of a presented "correction", and the behavioral and linguistic context of the correction. To measure such cognitive processing, we exploit the fact that a discrepancy between the expected and the observed system behavior results in characteristic patterns of brain activity. This pattern is called Error Potential (ErrP) and can be measured using Electroencephalography (EEG) almost immediately after erroneous system feedback is presented following a user action. A typical ErrP can be measured at fronto-central positions and occurs in a window of about 150 ms to 600 ms after a stimulus, i.e., the presentation of a correction, with its most pronounced components being a negative peak around 250 ms and a positive peak around 350 ms [10]. The exact contour and latency of an ErrP varies with tasks and individuals [20]. Work on "cortically-coupled computing" [13, 34] shows that by online classification of EEG potentials, a system reaction can precede a conscious human reaction to a target stimuli. Thus,

ICMI '17, November 13–17, 2017, Glasgow, United Kingdom

ACM ISBN 978-1-4503-5543-8/17/11…$15.00
https://doi.org/10.1145/3136755.3136784

[1] we estimate this fraction as 23%, based on the the probability for mis-typing a single letter of 5% [1] and an average word length in English of 5.

identifying ErrPs enables the design of systems that actively recover from errors. All this suggests that this technology is an ideal candidate for dealing with erroneous auto-corrections.

The ability to detect errors of an auto-correction mechanism would enable the possibility to automatically amend an initial auto-correction, for example by replacing it with an alternative or by providing additional ones. In this paper, we concentrate on the *detection* of such errors of the auto-correction. For this purpose, we use EEG features to capture occurring ErrPs in combination with context information derived from the predictive text entry keyboard. We demonstrate for the first time the feasibility of background error correction through EEG and context information without user involvement. For this purpose, we describe the recorded data corpus, the developed classifiers, and the performed evaluation.

1 RELATED WORK

1.1 Correction Approaches for Text Entry

In the HCI literature, different cues have been used to correct text input errors for a variety of input methods, including language models [16], keypress timing [6], hand postures [15], accelerometer data for mobile text entry [14], uncertainties in terms of finger positions on a touch screen [40], gestures during eyes-free input [37], geometric pattern matching [22], multi-modal input [36], as well as a combination of spatial and language models [11]. Modern approaches often support both word completion and typing correction [3]. While many of these techniques make use of automatic approaches to correct errors, most related work focuses on the reduction of of user errors during text entry. Little is known about how users can recover from errors of the auto-correction and how this can again be supported automatically.

1.2 ErrP detection for BCIs

Unsurprisingly, the idea to use ErrP detection for correcting errors made by a text-entry system has been first introduced in the context of Brain-Computer Interfaces (BCIs), as the necessary equipment for EEG recordings is routinely used there. BCIs as input or control device suffer from far-from-perfect recognition rates. A standard technique to remedy this is to always repeat each input several times. This increases robustness but leads to a low transfer rate [23]. The detection of ErrPs enables an increase in accuracy and therefore the potential transfer rate. Combaz et al. [7] showed how they can detect ErrPs during operation of a P300 speller BCI. They suggested to use the second-best recognition result of the BCI in case of a detected ErrP and showed through simulations how this would improve performance. Spüler et al. [35] pursued a different approach and deleted the previously given input in case an ErrP was detected. They then prompted the user to repeat the input command. The authors showed how they could use an online ErrP classifier to significantly increase transfer rate. A similar approach was chosen by Schmidt et al. [33], who showed that they could reliably detect ErrPs online and demonstrated a significant increase of communication speed for their gaze-independent "Center Speller" BCI. The work by Margaux et al. [27] is a rare example of studies that do not only investigate objective criteria but also subjective responses to error-aware interfaces. For a P300 speller with second-best correction, the authors showed that most participants "reported a preference in favor of a spelling including automatic correction". However, the authors also noted large individual differences regarding the subjective evaluation of their error-aware BCI. Llera et al. [26] used ErrPs to adapt the weight parameters of a logistic regression model for BCI operation to better represent the (assumingly) misclassified trial. They used simulation and offline analysis of data from eight participants to show that this process improved classification accuracy.

1.3 ErrP classification in general HCI

The number of studies which transfer ErrP detection to other input modalities, such as gestures, is limited. Förster et al. [12] used classification of ErrPs during operation of a gesture recognition system to improve its performance by adaptation of the gesture recognizer. However, their system did not immediately react to the detected ErrPs by error correction. Instead, it focused on improving gesture recognition accuracy by selective online adaptation: gesture trials which were classified correctly, i.e., did not result in an ErrP, were added to the training data to train a personalized gesture recognizer. This addressed a challenge for unsupervised adaptation, namely that the addition of misclassified trials can result in performance degradation instead of improvement. Vi and Subramanian [39] proposed an ErrP recognition system based on a consumer-level EEG device. They performed person-dependent ErrP classification, using a test set of 80 trials of Flanker Tasks and achieved a classification accuracy of about 0.7. Using a simulation of ErrP classification with different error rates, they also showed that a non-perfect ErrP detection rate between 0.65 and 0.8 was already beneficial for the enhancement of interactive systems for detecting user errors in spatial selection with the Flick technique on a touch surface. The authors analyzed accuracy improvements by allowing manual corrections when an ErrP is detected, but did not analyze costs or other usability aspects. Putze et al. [29, 30] showed that ErrPs can be detected in gesture-based user interfaces and used them to provide a self-correcting input mechanism. The authors used a model-based simulation to investigate the effect of different correction strategies and validated its predictions in a user study.

This literature review shows that single-trial classification of ErrPs from EEG is feasible in HCI contexts and can lead to measurable benefits for the user. However, applying ErrP classification for the detection of auto-correction errors of a predictive text entry method is challenging and it is unclear whether existing results can be transferred. Several challenges are associated with ErrP classification in the auto-correction context: 1) Auto-correction feedback to the user is subtle and can easily be missed. 2) As an erroneous auto-correction may only differ slightly from the intended word, it may not always be obvious to the user if and when an error occurred. 3) On the other hand, in some instances, users might expect auto-correction errors prior to their actual occurrence, e.g., for known flaws of the correction dictionary. 4) Auto-correction feedback perception overlaps with typing of the next word as users usually type continuously. 5) The user is (typically) operating a mobile device with manual commands, generating a lot of motion artifacts in the EEG signal. For those reasons, it is important to investigate whether ErrP classification is still possible under these

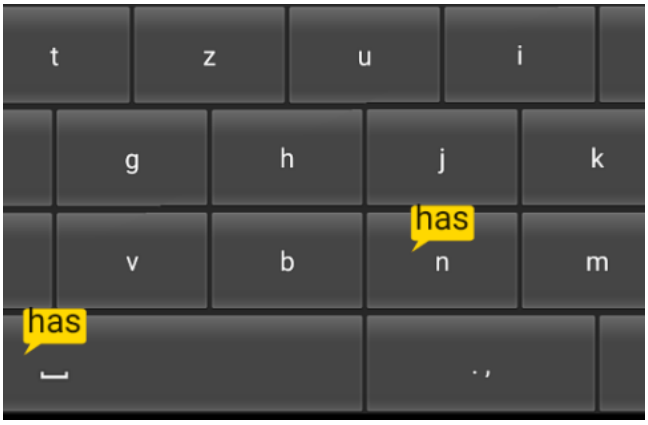

Figure 1: Screenshot of the custom keyboard. The notification (on the last pressed button and the space bar) indicates that a typed word was just replaced by an auto-correction.

challenging conditions and how the integration of context information can help in this context.

The rest of this paper is organized as follows: In the next three sections, we describe how we collected a set of EEG data from participants while they used a virtual keyboard with (sometimes incorrect) auto-corrections, we describe the classifier which we designed to detect auto-correction errors from this data, and we analyze the performance of the classification of auto-correction errors. The final section discusses the results and concludes the paper.

2 EXPERIMENTAL SETUP

2.1 Text Entry System

For collecting a data corpus of EEG data during text entry auto-correction, we designed and conducted an experiment. During the experiment, participants were asked to type short sentences presented to them on a virtual keyboard of a tablet computer. The keyboard contained a simple auto-correction mechanism which replaced erroneous, i.e., non-dictionary, words that the users typed with similar words from the dictionary. Brain activity during and after the presentation of those responses was recorded using EEG.

To present the target phrases, we used a slightly updated version of the TEMA software [4] on a Google Nexus 10 device running Android 6 as operating system. To ensure that participants generated and perceived an appropriate number of correct and wrong auto-corrections within a limited timeframe, we implemented a custom keyboard (based on the default Android keyboard layout) which was "rigged", in that it replaced typed letters with neighboring keys on the keyboard with a probability of 5%. The probability was chosen to mirror a typical typing error rate on soft keyboards [1]. This way, we could expect to record more data of correct and wrong auto-corrections within the allocated time, and gather sufficient data for training and testing of a classifier for auto-correction-associated ErrPs.

The implemented keyboard employs a simple correction mechanism: Corrections are restricted to full words after the participant pressed the space key or the enter key at the end of a sentence. Any entered non-dictionary word is replaced by a randomly chosen word from the set of dictionary entries with the lowest Levenshtein distance to the entered word. Such corrections do not trigger a suggestion, but an immediate replacement of the originally typed word. Due to ambiguity of the possible suggestions derived with this method, these replacements will contain errors. We extract EEG data and context information following each correction attempt and classify it to identify ErrPs indicating wrong corrections. It is important to differentiate between ErrPs resulting on the one hand from user errors (or errors induced by the rigged keyboard) and system errors resulting from the correction mechanism on the other hand. In the chosen design, each correction attempt is due to an (potentially rigged) entry error, as corrections are only performed for words which are not contained in the dictionary.

An important aspect of analyzing ErrPs is to ensure that the stimuli triggering the ErrP are regularly perceived. To maximize the probability of participants perceiving performed auto-corrections, the keyboard used four mechanisms: First, the target phrase to type was removed after the participant started typing to prevent participants from gazing at the text presentation and thus not be able to see potential corrections. Second, the word was replaced within the text field. Third, a sound was played and the tablet vibrated briefly for each correction. Fourth, the replacement was shown as a notification on the keyboard over the letter which was pressed last, as well as the space bar (see Figure 1). The last mechanism was introduced to make sure users noticed auto-corrections even when they focused their gaze on the keyboard instead of the typed text. The notification was displayed at multiple locations to maximize the probability that it appeared at a location at which the user is already looking. In contrast to a solution with notifications in a fixed location, e.g., in the center of the screen, our design allows users to naturally control their gaze. When evaluating ErrPs, we are only using EEG data from a segment following this notification, i.e., we do not expect many eye movement related artifacts to be superimposed on neural activity.

2.2 Data Collection

The phrase list from which phrases were presented randomly to the participants was extracted from the German OpenSubtitles phrase set [32]. We used only sentences without umlauts, converted text to lower case and removed all punctuation marks to avoid switching between different keyboard modes. Participants were asked to not use the backspace button to avoid interference between retroactive user activity and true correction events. Participants were told to write the sentences as fast and accurately as possible (favoring speed over accuracy if necessary). Furthermore, we instructed the participants to pay close attention to the performed corrections, allegedly because we were interested in their opinion on the correction quality.

In total, 12 university students and research assistants (seven female) participated in this experiment. Participants were aged between 19 and 36 years (average 26.0, standard deviation 5.1). All participants were informed about the nature of experiment and gave their written consent of participation. Participants received monetary compensation for their participation (10 Euro). Each

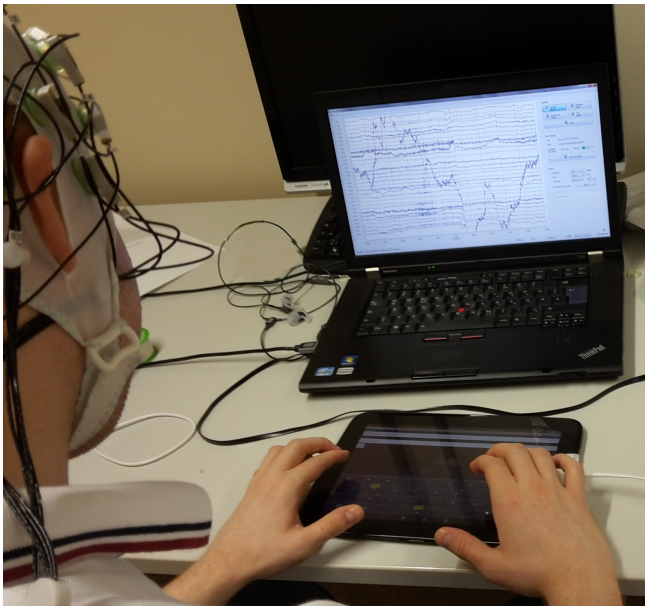

Figure 2: Photo of the experimental setup. Participant is wearing an EEG cap while typing on the table. Note that in the actual experiment, the participant could not see the EEG signal.

participant was given 15 training sentences to get accustomed to the keyboard, the implemented auto-correction and the overall flow of the experiment. After that, they typed 120 sentences. On average, this took 23.4 minutes (with a standard deviation of 4.8 minutes). After the experiment, participants were asked to fill a short questionnaire with 5-point Likert-scale items to assess how they judged their own performance and the performance of the system.

During the experiment, participants were equipped with an BrainProducts actiCAP with 32 active EEG electrodes. The electrodes were organized in a standard 32 channel actiCAP layout following the international 10-20 system. Pz was used as reference electrode. Impedance was kept below $16k\Omega$. Two electrodes were placed below and right to the right eye as EOG electrodes to capture ocular activity. The measurements were amplified by the actiCHamp amplifier and recorded via the PyCorder software. To synchronize the correction events with the EEG signal, the keyboard provided a color-switching box to which a light sensor of the recording setup was attached. Figure 3 shows a participant wearing the EEG headset while operating the virtual keyboard on the tablet.

3 DETECTION OF AUTO-CORRECTION ERRORS

We treat the task of detecting incorrect auto-corrections as a two-class classification problem. We extract EEG windows aligned to the presentation of the correction in the user interface, following the implicit assumption that the correction is perceived within a reasonably stable time frame after presentation. The `noError` class is assigned to windows with corrections which yielded the expected word, while the `error` class is assigned to windows corresponding to wrong corrections.

3.1 EEG and Context Features

For classification of ErrPs induced by incorrect first-order corrections, we combine features extracted from the EEG segment corresponding to the cognitive processing of the correction with context features derived from the correction procedure. Each window is baseline-corrected by subtracting the mean of the signal 200 ms before the window. Then, for each window and electrode position, we calculate two types of features: time-domain based and frequency-based ones. To calculate time-domain features, we partitioned each data window into smaller segments of 50 ms length. We then used the signal mean of the segment, calculated on the band-pass filtered signal, with cutoff frequencies at 4 and 13 Hz (i.e. $\theta-$ and $\alpha-$bands). For the frequency-based features, we calculated the frequency power spectrum between 4 and 13 Hz of the window, using Welch's method [41]. Those cutoff frequencies for filtering and frequency feature extraction were chosen because ErrPs are known to occur especially in the θ-band (see for example Cavanagh [5]); however, as frequency band borders are known to be person-dependent [9], we chose to include the α-band as well. As ErrPs can be expected to be most prominent at central-midline electrode positions, we restrict our feature extraction to the electrode positions Fz, Cz, and Pz. All features calculated on one window were stacked to form the tentative feature vector corresponding to that window.

In addition to the EEG features, we also derive several context features associated with each window. Those features encode information about the likelihood of a wrong correction and other parameters which might influence the generated EEG. The context features are as follows:

- **C1** Typing speed for replaced word relative to average typing speed
- **C2** Length of the replaced word (in number of characters)
- **C3** Time before user continues typing during evaluated EEG window (in ms)
- **C4** $1/N$, where N = number of candidate words of minimal Levenshtein distance to typed word

Feature C2 and C4 encode information about the likelihood of a wrong correction, based on the assumption that such an event is more likely when more words for replacement are plausible, which can be directly measured by the number of replacement candidates and indirectly measured by the length of the word. Features C1 and C3 encode information about the likelihood to actually notice the wrong replacement. We expect participants who type very swiftly and do not pause after a correction has occured, to more likely miss such an event.

Depending on the selection of tuning parameters, the described process generates thousands of features per window. As the resulting feature-space is large compared to the number of available training windows, we performed a feature selection using the Fisher ratio as selection criterion. The number k of selected features, i.e., those which exhibit the highest Fisher ratio, was a tuning parameter in the range between 5 and 50.

3.2 Classification of Imbalanced Data

For classification, we employed a Linear Discriminant Analysis (LDA) with co-variance shrinkage as regularization method. The shrinkage parameter was automatically determined using the analytic method by Ledoid-Wolf [24]. The LDA is trained in a person-dependent way, i.e., a new classification model is trained for each participant from his or her data.

A highly imbalanced class distribution as present in the PTE classification task may lead to degenerate classifiers with a strong bias towards the majority class (the `noError` class in our case) [2]. To cope with this challenge during training, we combine oversampling of the minority class with undersampling of the majority class. This processing chain is inspired by Putze et al.'s approach [31]. For oversampling, we employ the ADASYN algorithm [18]. ADASYN generates artificial samples by creating random interpolates between data points of the minority class. During this process, the algorithm generates most new samples from data points of the minority class which lie close to data points of the majority class, i.e. in regions which are critical for classification. We use ADASYN to generate additional training data for the `error` class. For undersampling of the majority class, we use a simple bagging approach in which we train several classifiers on randomly selected subsets of the majority class data (and all data of the minority class). Sampling is performed without replacement and set to a target class ratio of 0.7. ADASYN then generates synthetic samples to achieve a balanced class distribution in each training set. For each window, we perform a voting between all trained classifiers in the ensemble and assign the label `error` when the number of classifiers voting for this class exceeds a threshold. The number of weak classifiers within the ensemble is fixed to 100 in our setup.

The full classification pipeline is implemented in Python. For EEG processing, we use the MNE toolbox [17]. For machine learning and evaluation algorithms, we use scikit [28] and custom routines build on numpy and scipy. The methods for over- and undersampling are taken from the imbalanced-learn toolbox [25].

4 EVALUATION

4.1 Feature Analysis

Using the described experimental setup, we collected a total of 12 data sets. Each data set contained 120 sentences with a mean total number of 474.25 words (standard deviation of 6.3). Before the analysis, we excluded sentences in which the number of typed words did not match the number of expected words, i.e., with extra or omitted white space. This was done to avoid ambiguity about the alignment of the EEG window and the calculation of context features. Of all typed words, 25.85% were corrected (standard deviation of 3.7%). Of those corrections, 74.15% yielded the correct word (standard deviation of 13.1%). This indicates that the system must deal with an imbalanced classification problem, in which the `error` class is the minority class.

We analyze the responses to the post-experiment questionnaire to study typing behavior and possible implications. Table 2 shows the averaged agreement scores for all questions. Participants responded that they mostly looked at the keyboard during the experiment (agreement of 4.4 on a 5-point Likert scale with 5 corresponding to full agreement), which underlines the importance of correction feedback on the last-pressed key. Participants also reported that the given sentences were easy to memorize (agreement of 4.7), making it likely that auto-corrections could be checked for correctness. However, there was only medium agreement to the statement that the participant consistently perceived the correctness of the replacements (agreement of 3.7). This shows the challenging nature of classification of auto-correction errors as we can expect to encounter substantial label noise in cases where participants missed a wrong correction and did not experience an ErrP.

Table 1: Mean and standard deviation (in parentheses) for the context features, separated for `error` and `noError` class. Feature C1 is normalized by average typing speed.

	`noError`	`error`
C1	-2.8 (121.6)	-29.9 (128.1)
C2	5.5 (1.6)	4.1 (1.9)
C3	1646.8 (1525.7)	1875.1 (2960.0)
C4	0.9 (0.2)	0.4 (0.3)

Figure 4 exemplarily shows a Grand Average plot of the recorded EEG data for one participant, presenting the mean difference in signal amplitude for classes `error` and `noError`. We see that the EEG signal indeed reacts differently to stimuli of both classes, exhibiting strong deviations between 500 ms and 900 ms. We note that in comparison to more controlled recordings of ErrPs, the latency of the neural response in relation to the stimulus onset is larger. We explain this by considerable delay in stimulus perception, which requires the participant to note the correction attempt, process the result of the correction attempt, and detect the mismatch to expectation. This explanation is also consistent with the considerable delay until the next word is typed after a correction, see Table 1.

Table 1 summarizes statistics for the context features for classes `error` and `noError`. We see that these features exhibit significant differences in mean for the two classes, which makes them promising candidates for classification. For example, as expected, errors were more likely for shorter words (C2) and when fewer strong alternative corrections were available (C4). We also see that in case of an auto-correction error, users type slower (C1) and make slightly larger pauses afterwards (C4). However, considering the standard deviation in relation to the means (consider for example feature C1 – typing speed – from which a number of extreme outliers were already removed to calculate the mean in Table 1), we should expect substantial variation on a single-trial level, meaning that relying only on context features is likely not robust enough.

4.2 Classifier Evaluation

The classifier is evaluated in a 10-fold cross-validation. As evaluation metrics, we report classification accuracy, precision, recall, and F1-score (of the `error` class). Classification is performed for each participant individually and results for all participants are averaged. In Table 3, we summarize the results of the classifier of auto-correction errors. Furthermore, we report baseline results for the accuracy and F1-score metric. Baseline accuracy was determined as the relative frequency of the majority class (`noError`),

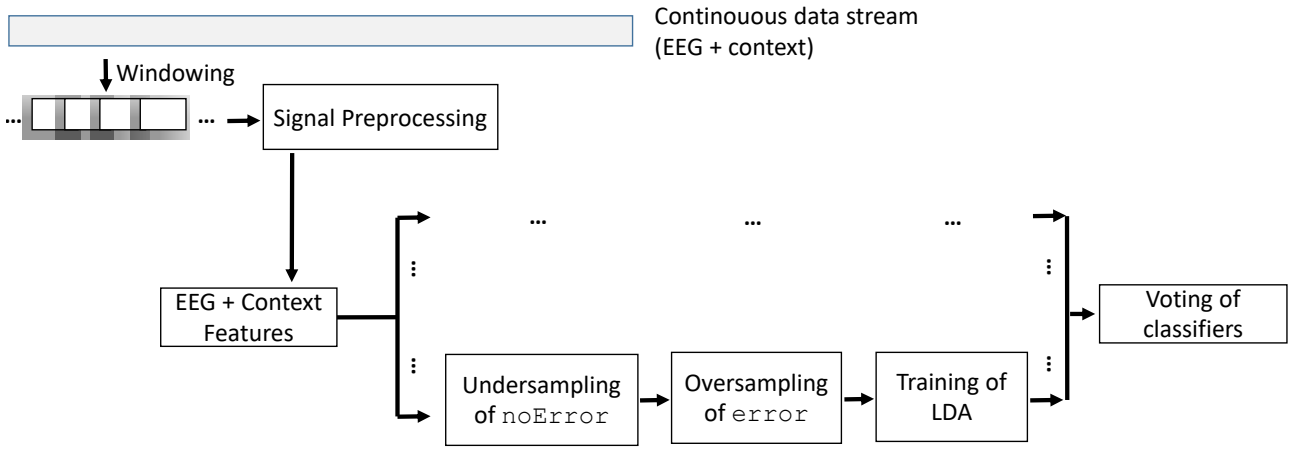

Figure 3: Schematic of the over- and undersampling approach for classification of auto-correction errors.

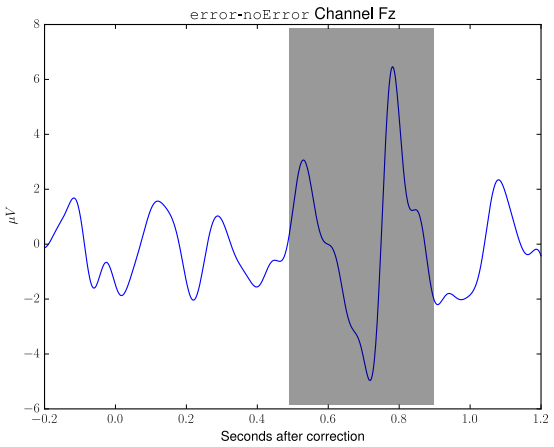

Figure 4: Grand Average (error-noError) of EEG data of a participant at electrode position Fz.

Table 2: Mean and standard deviation for the questionnaire results on text entry setup (AC = auto-correction). I...

	Mean	Std. Dev.
...made many mistakes	3.25	0.87
...wrote fluently	3.33	1.15
...noticed keyboard errors	3.25	1.71
...memorized prompts easily	4.67	0.49
...noticed whether AC correct	3.67	1.07
...expected occurence of AC	3.92	0.90
...looked at keyboard	4.42	0.67
...ignored AC	2.67	1.44
...noticed rigged letters	3.92	1.68
...found AC sometimes inexplicable	3.42	1.00
...could guess correctness of AC	2.5	1.00

while baseline F1-score was determined assuming a classifier which consistently predicts the target class (error), resulting in an assumed recall of 1.0 and an assumed precion equal to the relative frequency of the target class. This approach maximizes the baseline scores to which we compare to. As multiple components of the classification setup rely on non-deterministic processes, i.e., both over- and undersampling, we repeated the analysis for each participant 10 times and report results averaged across all repetitions. Table 3 shows that all performance metrics of the classifier indicate reasonable classification performance. Indeed, a paired, one-sided t-tests performed on the participant-wise F1-scores and accuracies shows that the classifier exhibits a performance which is significantly better than the respective baseline score ($p < 0.05$). This indicates that auto-correction error classification is feasible despite the challenging conditions. Our results indicate that the classifier as implemented is tuned more towards a high precision than towards high recall. For text entry applications, this is a desirable result as false positive results will yield distracting system actions when undoing correct auto-corrections. In contrast, false negatives will require the user to manually override the auto-correction, which is needed in current predictive text entry systems anyway. For other application scenarios, a different trade-off between precision and recall can be achieved through adjusting the parameters of the over- and undersampling procedure. We also employed paired, one-sided t-tests to show that the fusion classifier combining EEG and context features results performs significantly better than the two individual ones ($p < 0.05$ in both cases).

4.3 Feature Selection

The analysis of selected features gives us the opportunity to investigate feature stability both for individual and across sessions. Feature stability is an indicator of how robust the resulting classifiers are against variations in training and testing data. For individual sessions, we measure feature stability by counting the occurrence of features across cross-validation folds and bagging instances. As in each fold and for each bagging instance the training data is different, features which are selected consistently are robust against

Table 3: Classification auto-correction error results. We present mean and standard deviation for different feature combinations.

		Accuracy	Precision	Recall	F1 Score
a) Baseline	Mean	0.76	-	-	0.38
b) EEG	Mean	0.69	0.25	0.38	0.30
	Std. Dev.	0.08	0.10	0.13	0.07
c) Context	Mean	0.81	0.40	0.70	0.49
	Std. Dev.	0.03	0.19	0.25	0.18
d) Combined	Mean	0.85	0.82	0.65	0.72
	Std. Dev.	0.03	0.05	0.11	0.08

variations in the data. This analysis of feature counts reveals that averaged across all participants, 6.1 features are selected in more than 75% of all instances. This indicates the availability of a relatively stable feature set for each participant. We can also average feature counts across participants to analyze inter-personal stability. In this analysis, we see that 16 features were selected in more than 40% of instances (see Table 4 for the ten most frequently selected features). These results show a reasonable agreement across participants. As one would expect, cross-participant agreement on features is less than within-participant agreement due to individual differences in neural responses.

To understand which features are most relevant for the classification, we also look at which features were selected most often. Table 4 gives a summary of the ten most frequently selected features across all participants. We see that both spectral and time-domain features contribute to the selection. From the three employed electrodes, Fz and Cz dominate the selection, while Pz plays a minor role, as expected from the fronto-central localization of error potentials. To test if classification performance was due to task-related ocular artifacts in the EEG signal, we repeated the classification with added features from the EOG electrodes, which most clearly capture ocular activity. As the addition of these features did not improve classification accuracy, we conclude that the developed classification model does not rely on ocular artifacts.

5 DISCUSSION & CONCLUSION

In this paper, we described an experiment for collecting EEG data on auto-correction errors and showed that, despite challenging conditions, classification of auto-correction errors from EEG signals and context features is feasible and yields classification performance significantly better than the baseline. The combination of both types of features yielded a significant improvement over using any single one of the two feature types. While the EEG features alone were not discriminative enough, adding the context features allowed the classifier to pre-filter likely auto-correction errors. Furthermore, we analyzed the most frequently selected features and showed that both context features and a consistent set of EEG features contributed to the result. The benefit of the EEG features is significant but modest in effect size compared to the system only using context features. The improvement by EEG features also varied between participants, between 7% and 30% absolute. This shows that valuable information is encoded in the EEG data but that context information is necessary to uncover it.

Table 4: Relative frequency of the ten most frequently selected EEG features, indicating electrode positions and feature characteristic (central frequency for spectral features, segment center for time domain features).

Position	Feature	Rel. frequency
Cz	11.7 Hz	56.6%
Cz	0 s	49.5%
Fz	3.9 Hz	48.7%
Cz	9.7 Hz	48.2%
Cz	0.75 s	47.8%
Cz	0.25 s	47.0%
Cz	0.85 s	44.4%
Pz	0.45 s	44.3%
Cz	0.55 s	44.2%
Pz	11.7 Hz	44.1%

5.1 Limitations

As one of the first studies on the detection auto-correction errors, we purposefully limited the ecologically validity of the experimental design in exchange for increased controllability of recorded data. In future studies, ecologically validity of the employed experimental paradigm can be increased in several ways: First, while not all auto-corrections shown to the participants where induced by the rigged keyboard, the majority (90%) of them was. Errors induced by actual user errors are likely different from keyboard-induced errors. For example, while asking users to type predefined, memorized phrases is an established paradigm in text entry research [38], this potentially limits the validity of the presented results in comparison to unrestricted text entry of own text. The same holds for the restriction not to correct typing errors. While allowing such unrestricted text entry would diversify the types of auto-correction errors and their distribution (e.g., we would see repeated errors for out-of-dictionary words which are always "corrected" to a wrong word), it would also provide additional context features for classification (e.g., by taking usage of backspace into account).

We see a lot of potential for improving the EEG-based auto-correction error detection in future work. While the classifier used context features, it was based on a simple auto-correction mechanism based on much less information compared to what is available to more sophisticated auto-correction methods. Additionally, apart from standard signal processing, we did not use any techniques to remove artifacts or other confounding influences from the EEG signals. As participants moved while typing, the generated artifacts likely influenced the performance of the classifier in a negative manner. Identifying or removing such artifacts could improve classification accuracy. This would be especially relevant if we further consider truly mobile usage. De Vos et al. [8] showed that while performance of a BCI during outdoor walking decreases compared to a desktop situation, it still provides useful results without recalibration.

From an HCI perspective, it is clear that users in many text entry scenarios will not tolerate wearing an EEG cap for an improvement in auto-correction accuracy. However, the last decade has seen a rise of alternative EEG devices beyond traditional caps, either with the goal of providing off-the-shelf EEG acquisition (such as the emotiv Epoc or the NeuroSky headset) or with the goal to provide flexible, unobtrusive EEG acquisition for researchers or hackers (such as cEEGrid or OpenBCI). Dry electrodes which do not require electrode gel are available from several manufacturers. While these devices do not yet provide the signal quality of traditional cap systems, they are portable, comfortable, and affordable. The first applications which may benefit from the presented detection of auto-correction errors are in the domain of professional text entry, where errors are costly or even dangerous.

5.2 Future Work

To further improve the classification performance, we will extend the available feature set in future studies. For this purpose, we will add more context features, such as dynamic typing speed measures, or n-gram statistics of replacement candidates. Furthermore, we will extend the available information sources with eye tracking data which will allow us to achieve a better alignment of classification windows and provide features which encode eye gaze fixations on displayed auto-corrections.

We are also planning to integrate the detection of auto-correction errors in text entry applications. The main application we envision is the implementation of *second-order correction* approaches which actively recover from detected auto-correction errors by suggesting alternative corrections. A correction strategy might be to respond to a detected auto-correction error depending on the confidence of the original correction: If confidence is low, the original auto-correction is automatically replaced by the second-best one; if confidence is high (i.e., a false alarm is likely), a selection of alternatives is suggested to the user. A first step towards an evaluation of such a strategy will be the simulation of multiple strategies to estimate the effect of different auto-correction contexts compared to manual correction. Afterwards, an online system will be implemented to perform actual user tests. The key challenge for an online system is to reduce the amount of necessary calibration data per person. The current system uses the full available data of each person (minus the testing data of the respective crossvalidation folds) for person-dependent training. It should be noted that due to the undersampling step of the balancing mechanism, not all of that data is actually used for training, i.e., by enforcing an equal distribution of the classes during training, the amount of training samples could be reduced by 30%. Transfer learning approaches [21] could be used to further reduce the number of required training episodes.

REFERENCES

[1] A. S. Arif and W. Stuerzlinger. 2009. Analysis of text entry performance metrics. In *Science and Technology for Humanity (TIC-STH), 2009 IEEE Toronto International Conference*. 100–105.

[2] Rukshan Batuwita and Vasile Palade. 2013. Class Imbalance Learning Methods for Support Vector Machines. In *Imbalanced Learning*, Haibo He and Yunqian (Eds.). John Wiley & Sons, Inc., 83–99.

[3] Xiaojun Bi, Tom Ouyang, and Shumin Zhai. 2014. Both Complete and Correct?: Multi-objective Optimization of Touchscreen Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2297–2306.

[4] Steven J. Castellucci and I. Scott MacKenzie. 2011. Gathering Text Entry Metrics on Android Devices. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems (CHI EA '11)*. ACM, New York, NY, USA, 1507–1512.

[5] James F Cavanagh, Michael J Frank, Theresa J Klein, and John JB Allen. 2010. Frontal theta links prediction errors to behavioral adaptation in reinforcement learning. *Neuroimage* 49, 4 (2010), 3198–3209.

[6] James Clawson, Kent Lyons, Alex Rudnick, Robert A. Iannucci, Jr., and Thad Starner. 2008. Automatic Whiteout++: Correcting mini-QWERTY Typing Errors Using Keypress Timing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 573–582.

[7] Adrien Combaz, Nikolay Chumerin, Nikolay V. Manyakov, Arne Robben, Johan A. K. Suykens, and Marie M. Van Hulle. 2012. Towards the detection of error-related potentials and its integration in the context of a P300 speller brain–computer interface. *Neurocomputing* 80 (2012), 73–82.

[8] Maarten De Vos, Katharina Gandras, and Stefan Debener. 2014-01-01. Towards a truly mobile auditory brain–computer interface: Exploring the P300 to take away. 91, 1 (2014-01-01), 46–53.

[9] Michael Doppelmayr, Wolfgang Klimesch, Th Pachinger, and B Ripper. 1998. Individual differences in brain dynamics: important implications for the calculation of event-related band power. *Biological cybernetics* 79, 1 (1998), 49–57.

[10] Pierre W. Ferrez and José del R. Millan. 2008. Error-Related EEG Potentials Generated During Simulated Brain Computer Interaction. *IEEE Transactions on Biomedical Engineering* 55, 3 (2008), 923–929.

[11] Andrew Fowler, Kurt Partridge, Ciprian Chelba, Xiaojun Bi, Tom Ouyang, and Shumin Zhai. 2015. Effects of Language Modeling and Its Personalization on Touchscreen Typing Performance. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 649–658.

[12] Kilian Förster, Andrea Biasiucci, Ricardo Chavarriaga, Jose del R Millan, Daniel Roggen, and Gerhard Tröster. 2010. On the Use of Brain Decoded Signals for Online User Adaptive Gesture Recognition Systems. In *Pervasive Computing*. Number 6030 in Lecture Notes in Computer Science. Springer Berlin Heidelberg, 427–444.

[13] A.D. Gerson, L.C. Parra, and P. Sajda. 2006. Cortically coupled computer vision for rapid image search. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 14, 2 (June 2006), 174–179.

[14] Mayank Goel, Leah Findlater, and Jacob Wobbrock. 2012. WalkType: Using Accelerometer Data to Accomodate Situational Impairments in Mobile Touch Screen Text Entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 2687–2696.

[15] Mayank Goel, Alex Jansen, Travis Mandel, Shwetak N. Patel, and Jacob O. Wobbrock. 2013. ContextType: Using Hand Posture Information to Improve Mobile Touch Screen Text Entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2795–2798.

[16] Joshua Goodman, Gina Venolia, Keith Steury, and Chauncey Parker. 2002. Language Modeling for Soft Keyboards. In *Proceedings of the 7th International Conference on Intelligent User Interfaces (IUI '02)*. ACM, New York, NY, USA, 194–195.

[17] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A Engemann, Daniel Strohmeier, Christian Brodbeck, Roman Goj, Mainak Jas, Teon Brooks, Lauri Parkkonen, and others. 2013. MEG and EEG data analysis with MNE-Python. *Frontiers in neuroscience* 7 (2013), 267.

[18] Haibo He, Yang Bai, E. A. Garcia, and Shutao Li. 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. 1322–1328.

[19] Christopher J. Hynes. 2016. Device, Method, and Graphical User Interface for Visible and Interactive Corrected Content. (April 2016).

[20] Inaki Iturrate, Ricardo Chavarriaga, Luis Montesano, Javier Minguez, and Jose del R Millan. 2012. Latency correction of error potentials between different experiments reduces calibration time for single-trial classification. *Proceedings of Annual International Conference of the Engineering in Medicine and Biology Society.* 2012 (2012), 3288–3291.

[21] Vinay Jayaram, Morteza Alamgir, Yasemin Altun, Bernhard Scholkopf, and Moritz Grosse-Wentrup. 2016. Transfer learning in brain-computer interfaces. 11, 1 (2016), 20–31.

[22] Per-Ola Kristensson and Shumin Zhai. 2005. Relaxing Stylus Typing Precision by Geometric Pattern Matching. In *Proceedings of the 10th International Conference on Intelligent User Interfaces (IUI '05)*. ACM, New York, NY, USA, 151–158.

[23] Dean J. Krusienski, Eric W. Sellers, Dennis J. McFarland, Theresa M. Vaughan, and Jonathan R. Wolpaw. 2008. Toward enhanced P300 speller performance. *Journal of Neuroscience Methods* 167, 1 (2008), 15–21.

[24] Olivier Ledoit and Michael Wolf. 2003. *Honey, I Shrunk the Sample Covariance Matrix.* SSRN Scholarly Paper ID 433840. Social Science Research Network, Rochester, NY.

[25] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. 2017. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research* 18, 17 (2017), 1–5.

[26] Andreas Llera, Marcel A. J. van Gerven, Victor M. Gómez, Ole K. Jensen, and Hilbert J. Kappen. 2011. On the use of interaction error potentials for adaptive brain computer interfaces. *Neural Networks* 24, 10 (2011), 1120–1127.

[27] Perrin Margaux, Maby Emmanuel, Daligault Sébastien, Bertrand Olivier, and Mattout Jérémie. 2012. Objective and Subjective Evaluation of Online Error Correction During P300-based Spelling. *Adv. in Hum.-Comp. Int.* 2012 (2012).

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[29] Felix Putze, Christoph Amma, and Tanja Schultz. 2015. Design and Evaluation of a Self-Correcting Gesture Interface Based on Error Potentials from EEG. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 3375–3384.

[30] Felix Putze, Dominic Heger, and Tanja Schultz. 2013. Reliable subject-adapted recognition of EEG error potentials using limited calibration data. In *6th International Conference on Neural Engineering*. San Diego, USA.

[31] Felix Putze, Johannes Popp, Jutta Hild, Jürgen Beyerer, and Tanja Schultz. 2016. Intervention-free selection using EEG and eye tracking. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*. ACM, 153–160.

[32] Germán Sanchis-Trilles and Luis A. Leiva. 2014. A Systematic Comparison of 3 Phrase Sampling Methods for Text Entry Experiments in 10 Languages. In *Proceedings of the international conference on Human-computer interaction with mobile devices and services (MobileHCI)*.

[33] Nico M. Schmidt, Benjamin Blankertz, and Matthias S. Treder. 2012. Online detection of error-related potentials boosts the performance of mental typewriters. *BMC Neuroscience* 13 (2012), 13–19.

[34] Pradeep Shenoy and Desney S. Tan. 2008. Human-aided Computing: Utilizing Implicit Human Processing to Classify Images. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 845–854.

[35] Martin Spüler, Michael Bensch, Sonja Kleih, Wolfgang Rosenstiel, Martin Bogdan, and Andrea Kübler. 2012. Online use of error-related potentials in healthy users and people with severe motor impairment increases performance of a P300-BCI. *Clinical neurophysiology: official journal of the International Federation of Clinical Neurophysiology* 123, 7 (2012), 1328–1337.

[36] Bernhard Suhm, Brad Myers, and Alex Waibel. 2001. Multimodal Error Correction for Speech User Interfaces. *ACM Trans. Comput.-Hum. Interact.* 8, 1 (March 2001), 60–98.

[37] Hussain Tinwala and I. Scott MacKenzie. 2010. Eyes-free Text Entry with Error Correction on Touchscreen Mobile Devices. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries (NordiCHI '10)*. ACM, New York, NY, USA, 511–520.

[38] Keith Vertanen and Per Ola Kristensson. 2014. Complementing text entry evaluations with a composition task. 21, 2 (2014), 8.

[39] Chi Vi and Sriram Subramanian. 2012. Detecting error-related negativity for interaction design. In *Proceedings of the Conference on Human Factors in Computing Systems*. New York, USA.

[40] Daryl Weir, Henning Pohl, Simon Rogers, Keith Vertanen, and Per Ola Kristensson. 2014. Uncertain Text Entry on Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2307–2316.

[41] Peter D. Welch. 1967. The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *Transactions on Audio and Electroacoustics* 15, 2 (1967), 70–73.