A Multi-Context Character Prediction Model for a Brain-Computer Interface

Shiran Dudy and **Steven Bedrick**
Center for Spoken Language Understanding
Oregon Health & Science University
Portland, OR, USA
{dudy,bedricks}@ohsu.edu

Shaobin Xu and **David A. Smith**
College of Computer and Information Sciences
Northeastern University
Boston, MA, USA
{shaobinx,dasmith}@ccs.neu.edu

Abstract

Brain-computer interfaces and other augmentative and alternative communication devices introduce language-modeing challenges distinct from other character-entry methods. In particular, the acquired signal of the EEG (electroencephalogram) signal is noisier, which, in turn, makes the user intent harder to decipher. In order to adapt to this condition, we propose to maintain ambiguous history for every time step, and to employ, apart from the character language model, word information to produce a more robust prediction system. We present preliminary results that compare this proposed Online-Context Language Model (OCLM) to current algorithms that are used in this type of setting. Evaluations on both perplexity and predictive accuracy demonstrate promising results when dealing with ambiguous histories in order to provide to the front end a distribution of the next character the user might type.

1 Introduction

Augmentative and alternative communication (AAC) devices are aimed at individuals facing communication disabilities, and aim to enable them to interact with their environment, assisting with both comprehension as well as expression of the individual (Beukelman and Mirenda, 2005; American Speech Language Hearing Association et al., 2004). In particular, a Brain-Computer Interface (BCI) has been employed as an aiding device for patients who have experienced loss of motor control, and who might struggle with producing spoken or written language (Birbaumer et al., 1999; Sellers et al., 2010) (as in e.g. Amyotrophic Lateral Sclerosis). A BCI system measures a user's brain's electrical activity, typically using electroencephalography (EEG), and attempts to infer intent, often in response to stimuli such as row-column scanning.

Communication rates in BCI systems tend to be quite slow. To reduce the prediction time of the next letter in the sequence, one approach is to incorporate language models into AAC devices (Vertanen and Kristensson, 2011), and BCI in particular (Mora-Cortes et al., 2014). The potential advantage of combining a language model with EEG information in a BCI system is twofold: achieving higher accuracy of target predictions, as well as gains in the speed of the process of typing the user's intended string (Speier et al., 2011; Orhan et al., 2011). It may also help with reducing the number of necessary stimuli required for symbol selection (Speier et al., 2016).

Mora-Cortes et al. (2014) describe different types of EEG responses common when employing a typing-based BCI, and in the review of Speier et al. (2016), event-related potentials (ERP) seem to be the prevailing choice in particular. In the RSVP-Keyboard (Orhan et al., 2012), an ERP signal is elicited in response to a rapid display of letters. Other systems such as the P300 (Farwell and Donchin, 1988) present a user with a grid of letters and the ERP signal is retrieved in response to a flash of each row or column.

In such systems, a language model provides the prior distribution that serves as a bias to the EEG evidence when computing the posterior distribution of a symbol as shown in Equation 1

$$p(sym|EEG) = \frac{p(sym)p(EEG|sym)}{p(EEG)} \quad (1)$$

Traditionally, such BCI systems commit to a single decision (given their posterior results), which introduces a problem once the decision is not aligned with the user. One option is to allow for a backspace character, though this poses complex modeling challenges (Fowler et al., 2013). In our system we propose to partly commit to a decision with regard to the user intent, and maintain more than one candidate for every letter selection.

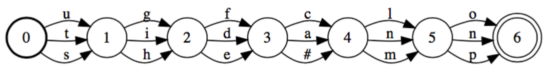

Figure 1: EEG evidence for 6 letter selections

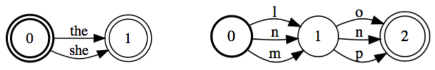

Figure 2: left to right *word history* and *trailing prefix*

This way, the best path is frequently re-evaluated and as a result can revise previously typed strings. Maintaining more than one candidate may reduce frustration among users, and minimize time spent correcting errors. We also, as recommended by Mora-Cortes et al. (2014) and Speier et al. (2016), explicitly incorporate word-level information and mix word-level and character-level language models to improve letter level prediction. Notably, this approach allows our system to support out-of-vocabulary (OOV) words.

Ambiguous input is widely used in traditional Automatic Speech Recognition (Jelinek, 1997) systems along with the Viterbi algorithm (Viterbi, 1967) to find the best path of the utterance bottom-up, from acoustic phoneme units (often triphones) to characters and words (often represented as the $C \circ L \circ G$ composition), and recently in keyboard settings as well by swiping (Ouyang et al., 2017). To the best of our knowledge, it has rarely been explored in the area of BCI, with a notable exception being the work of Speier et al. (2015), who later found it to be intractable due to complexity issues (Speier et al., 2016).

In this work, we present the *Online-Context Language Model* (OCLM) and describe the process of producing prior distributions given EEG evidence as part of our BCI system. Our contributions are specifically in the context of BCI as we integrate word language models (within the system), allow for OOV words to be typed, and optimize over word-level paths to compute optimal priors. Our mixed-context approach achieves superior performance in the face of noisy input compared with a purely character-level model.

2 Methodology

2.1 Overview

To illustrate our method, the steps presented are applied on the input in Figure 1, which depicts a temporal sequence of EEG evidence. Each numbered state represents a selection epoch in which a user has attempted to "type" a letter. Arcs transitioning between nodes i to $i + 1$ represent possible selections, and (though not drawn as such in this figure) are weighted with the normalized EEG likelihoods of time step i retrieved from the BCI. In this example, the OCLM is currently aiming to provide a prior distribution for the 7th letter selection given the EEG evidence. Assume that the intended string the user wishes to type is "the mo".

The first step taken is to split the EEG history lattice shown in Figure 1 into two parts: one, containing all strings before the last space ("#" in Figure 1), which we will call the "word history" lattice, and another, containing all possible strings *after* the last space, called the "*trailing prefix*" (F_{tp}) as shown in Figure 2.[1]

The *word history* (F_{wh}) represents all possible complete words that the user is assumed to have finished typing. The second part represents the word that the user is assumed to be in the middle of typing, and can be thought of as the set of possible prefixes for the current intended word. We can use this to generate possible word *completions*.

A current in-progress word lattice (F_{cw}) is generated by finding all possible in-vocabulary words that start with the "trailing prefix." By concatenating both the word history lattice (F_{wh}) as well as a lattice of possible in-progress words, and then composing the result with a word-level language model (F_{wLM}), we are able to produce a weighted n-best list of hypothesized words that the user may be trying to produce (F_{top_n}, Equation 2).

$$F_{top_n} = (F_{wh} \| F_{cw}) \circ F_{wLM} \tag{2}$$
$$F^{char}_{top_n} = proj_{char}(F_{top_n} \circ F_{spellout}) \tag{3}$$
$$F_{sym} = F^{char}_{top_n} \cup F_{charLM} \tag{4}$$
$$F_{OCLM} = (F_{tp} \| \sigma) \circ F_{sym} \tag{5}$$

We next take the lattice of "current words", compose that with a word-to-character spellout machine ($F_{spellout}$), project the result into character space (Equation 3), and then take the union of the result with a character language model to produce a "symbol language model" lattice (F_{sym} in Equation 4).

We are now ready to compute probabilities for the likely next symbol the user intends to select. We do this by taking our original trailing prefix

[1]The *trailing prefix* shown in figure 2 is simplified for illustrative purposes; in this example, it also includes other paths from state 0 that do *not* include "#". Note also that all paths contain the same number of characters since backspaces are omitted.

lattice (F_{tp}, containing distributions over characters actually selected by the user), concatenating it a character "wild card" machine (σ, representing the next character that the user intends to select to continue F_{tp}), and intersecting the resulting machine with F_{sym} (Equation 5). At this point, we are able to extract a probability distribution over possible character selections by performing weight pushing to the end of the machine, and then examining the final set of arcs in F_{OCLM}. All of these FST operations are implemented with the OpenFst (Allauzen et al., 2007) and OpenGRM-NGRAM (Roark et al., 2012) libraries.

2.2 Out of Vocabulary Words

It is crucial that our system allows its users to enter novel, out-of-vocabulary words, which we support in the following way. Above, we described how we take the lattice of characters making up the current, in-progress word and combine it with a lexicon transducer to produce a weighted lattice of possible word completions. Our approach deals differently and distinguishes between rarely seen words in the training set and completely unseen words. Infrequent words that were seen less than five times but at least once during training are mapped to <*unk*> word-symbol. In these cases, an attempt to complete a prefix for such a word results in an <*unk*> word token as a possibility in F_{cw}'s list. F_{wLM} is also trained on these <*unk*> representations suggesting that <*unk*> is treated similarly to any other word appearing in the training after the infrequent word was mapped to it (following the process in Equations 2 to 5). However, while the described process applies to uncommon but observed words, when a prefix is part of an *unseen* word and there is no possible completion provided by the lexicon then, $F_{cw} = \emptyset$ and the process is reduced to Equations 4 and 5 such that $F_{sym} = F_{charLM}$. The F_{charLM} contains a failure (ϕ) transition such that for every possible prefix in F_{tp} that is not found in F_{sym} (which in this situation is equivalent to F_{charLM}), we back off to a partial prefix route.

2.3 Word Completion

As described above, while computing the prior distribution over the next character, every new piece of evidence that is added to the system (i.e., each new attempt at character selection by the user) is appended to a "trailing prefix," F_{tp}, which we use to compute hypotheses about the current words the user might be in the middle of typing. In Equation 2 above, F_{cw} contains all possible in-vocabulary words that the user may be trying to produce, and F_{top_n} is a refined list of those words. This means that our architecture is able to produce both word- and character-level predictions simultaneously, and can easily vary the number of such predictions.

2.4 Auto-correction

While character deletion or insertion events are not auto-corrected in our approach, we address auto-correction to some degree. As described in the main process of computing the prior distribution, there is a frequent re-assessment the previously typed words. This history is recalculated as more EEG evidence of letter selections is introduced; this recalculation occurs mainly when the space character is introduced as it affects the word-boundaries and thus the history. As a result, at each time step, the updated history is rescored by the word language-model (F_{wLM}), as our estimated most-likely word history may have changed. In Figure 1, for instance, we start with "she" and "the" as possible histories. As character selection continues, we might have "monkey" as a strongly-predicted possible word completion; in this case, F_{wLM} will prioritize "the" over "she," as "the monkey" is more probable than "she monkey". However, if "modifies" were to be judged more likely by F_{top_n}, "she" would get prioritized over "the". Making additional and extended use of this information is part of our future work plan.

3 Experimental Evaluation

In order to evaluate the OCLM, we constructed a simulated character selection task to compare the performance of three different algorithms. The first is a smoothed character 5-gram model ("NGRAM"),[2] in which prediction is conducted by intersecting the history (EEG evidence, plus the wildcard σ) with the LM lattice. In this and our other n-gram models, we used Kneser-Ney smoothing (Kneser and Ney, 1995). The second algorithm we evaluated against was a "prefix language model" that included a character language model (PreLM, Equation 6) explicitly trained on character prefixes of in-vocabulary words (F_{wp})

[2]Our choice of an ngram order of 5 was for empirical reasons. A larger window would be too sparse and memory-intensive; a shorter window would not completely capture some of the shorter words that we wanted the model to learn.

combined with a closure over the NGRAM model. This model used an identical prediction mechanism as did the *NGRAM* model, but was designed to be better-equipped to handle in-progress words.

$$PreLM = (F_{NGRAM} \| F_{space})^* \| F_{wp} \quad (6)$$

The third is the OCLM described in Section 2. All models were trained on the Brown corpus (Francis and Kučera, 1979), and data split was 80% for training and 20% for testing. The results are on the test set. The test set contained $10,265$ sentences and $176,280$ words. Our simulated task was as follows. For each test sentence, we proceed character by character. At each point in the sentence, we provide the algorithms with some representation of the history of the sentence up to that point, and ask the models to predict the following character target.

We evaluated each model in terms of the traditional average character-level perplexity. Since our model is intended to be used to support character prediction, we also measured the reciprocal rank of the ground-truth target letter, both overall as well as by position within a word (to compare performance earlier in words vs. later). We also looked at the proportion of predictions for which the model placed the correct target word in the top 10 sorted guesses (ACC@10). We evaluated under two conditions: the first simulating a "deterministic" history, in which we assume that the EEG signal is reliable, and an "ambiguous" history, in which we take the top-*n* letters from the EEG signal at each time point as possible selections.

3.1 Deterministic History

In this condition, the language model was provided with the "correct" character history up to each prediction point. We refer to this as the n=1 scenario, as it describes the state of deterministic history of unambiguous and accurate EEG selection at each time point.

Table 1 shows that NGRAM algorithm has subpar performance when compared to both OCLM and PreLM as its Mean Reciprocal Rank (MRR) as well as Perplexity (PPX) are lower and higher respectively. ACC@10 that stands for sentence average prediction for the target within top 10 guesses also demonstrate inferior performance for the NGRAM method.

OCLM and PreLM appear to have similar performance, which surprised us given how different

metric	NGRAM	PreLM	OCLM
MRR	0.4	0.7	0.75
PPX	4.4	1.8	1.9
ACC@10	0.69	0.96	0.96

Table 1: Evaluation Results (n=1)

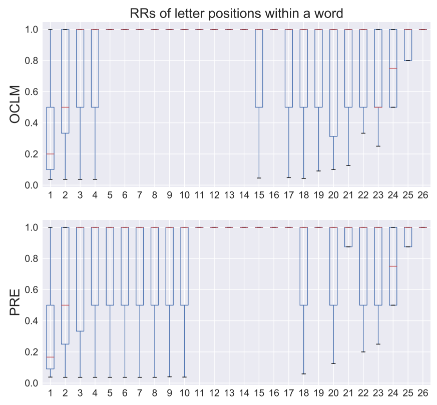

Figure 3: Reciprocal Ranks of letter positions within words (n=1)

their approaches are. In order to explore more deeply, we attempted to examine the internal behavior within each word in the test set. In Figure 3 the Reciprocal Ranks (RR), are presented for each letter position within a word. The OCLM algorithm demonstrates has higher RR (than PreLM) earlier in the word (position 5 vs. position 11) which is maintained for a longer period of time (10 positions in a row vs 7). Arguably, it is desired that the algorithm would predict correctly as early as possible, as its impact is on greater number of words (there are more words with 5 letters than with 10) and, a longer duration of high RR contributes to more accurate predictions as well. An analogous analysis of perplexity found a similar pattern of *lower* perplexity at *earlier* positions, together with similar duration, and a similar performance profile in OCLM vs PreLM.

3.2 Ambiguous History

Maintaining an ambiguous history of characters for each character selection has the potential to preserve routes the system might have missed otherwise. Erroneous decisions of the system with regards to character predictions unfortunately exist, especially given a noisy channel to process the data from such as the BCI system (different to some degree than typing with a keyboard). In this experiment we examined the performance of OCLM and PreLM with a simulated ambiguous history. We evaluated the OCLM and PreLM algorithms' performance when given histories consisting of the most n likely EEG characters at each time point for $n \in \{2, 3\}$. In all cases, the target character was included, but it was not always the most "probable." Likelihoods were drawn from a Gaussian PDF functions of target and non-target simulating a high performing user [3].

In BCI settings, utterances are often shorter than those that are common in datasets such as the Brown Corpus; therefore, we evaluated on a subset of the test-set containing all sentences of 10 words or less, resulting in $2,954$ sentences with $16,519$ words. For reasons of computational tractability, we simulated an ambiguous history over a window of the immediate past 10 characters; beyond that, we treated history as "fixed," and provided the models with correct targets.[4]

Table 2 presents the results for ambiguous input of n=2 and n=3. Overall adding more history to these type of algorithms degrades their performance. However, the level of degradation seems to vary between OCLM to PreLM. OCLM can produce more accurate predictions and has a higher MRR rate for both ns. PPX also is slightly lower than in PreLM.

nbest	metric	PreLM	OCLM
	MRR	0.29	0.51
n=2	PPX	3.5	3.0
	ACC@10	0.69	0.87
	MRR	0.26	0.44
n=3	PPX	4	3.9
	ACC@10	0.63	0.83

Table 2: Evaluation Results (n=2, n=3)

A further inspection[5] into the letter position performance of n=2 shows that OCLM's MRR remains relatively high (close to 1) especially from the fifth letter position while PreLM falls to values around 0.2. While not shown, the PPX values are consistent again with the MRR behavior. The same pattern appears with a noisier history of n=3. There, OCLM experienced additional degradation, and its high MRR lasted for a shorter term; from the seventh position to the tenth. PreLM's MRR revolved around slightly less than 0.2. OCLM's PPX was relatively low yet higher than of n=2 and was higher for PreLM. This, together with Table 2 indicate a performance *degradation* in OCLM, but a *break-down* in performance for PreLM. These results emphasize the durability of OCLM over PreLM when the input has more than one possibility.

[3]The PDFs were overlapping to some degree to enable likelihood confusions of target with non-target

[4]PreLM fails to run on complete ambiguous history

[5]Not included in this work for reasons of space.

4 Conclusions

We presented the OCLM architecture for predictive typing with a brain-computer interface. The OCLM enables incorporating ambiguous histories and word-level knowledge to improve its predictions. The OCLM demonstrated improved prediction quality as it takes place earlier in the process and for a longer duration across different conditions of ambiguous input to the system. Our architecture also allows for personalization by employing another lattice to Equation 4 of a user's probable words.

Our future work will focus on integrating the OCLM architecture into our group's BCI system, and evaluating the algorithm with real end-users to see if it reduces the number of sequence displays for each letter selection. Since speed is an important component in the usability of a BCI system, we also plan to assess total typing time as well as accuracy; note that, as our system's predictions become more accurate, the downstream components of our BCI system will also improve.

Another important area of future work will be to more thoroughly investigate our system's handling of OOV words, and identify avenues for improvement as needed. We also hope to take advantage of our model's architecture to provide user-level personalization, and to explore more concrete approaches to autocorrection. Our work to date has focused on the AAC literature, and there is current work in areas such as spelling correction that may prove useful here.

Acknowledgments

We would like to thank the reviewers of the SCLeM workshop for their insightful comments and feedback. We also would like to thank Brian Roark for his helpful advice, as well as our clinical team in the Institute on Development & Disability at OHSU. Research reported in this paper was supported the National Institute on Deafness and Other Communication Disorders of the NIH under award number 5R01DC009834-09. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. OpenFST: A General and Efficient Weighted Finite-State Transducer Library. In *International Conference on Implementation and Application of Automata*, pages 11–23. Springer.

American Speech Language Hearing Association et al. 2004. Roles and Responsibilities of Speech-Language Pathologists with Respect to Augmentative and Alternative Communication: Technical Report.

David R. Beukelman and Pat Mirenda. 2005. *Augmentative & Alternative Communication: Supporting Children & Adults with Complex Communication Needs*, 3rd edition. Paul H. Brookes Publishing Co.

Niels Birbaumer, Nimr Ghanayim, Thilo Hinterberger, Iver Iversen, Boris Kotchoubey, Andrea Kübler, Juri Perelmouter, Edward Taub, and Herta Flor. 1999. A Spelling Device for the Paralysed. *Nature*, 398(6725):297.

Lawrence Ashley Farwell and Emanuel Donchin. 1988. Talking Off the Top of Your Head: Toward a Mental Prosthesis Utilizing Event-Related Brain Potentials. *Electroencephalography and Clinical Neurophysiology*, 70(6):510–523.

Andrew Fowler, Brian Roark, Umut Orhan, Deniz Erdogmus, and Melanie Fried-Oken. 2013. Improved Inference and Autotyping in EEG-Based BCI Typing Systems. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '13, pages 15:1–15:8, New York, NY, USA. ACM.

WN Francis and H Kučera. 1979. Brown Corpus Manual of Information to Accompany a Standard Corpus of Present-Day American English, revised and amplified. *Providence, RI: Brown University, Department of Linguistics*.

Frederick Jelinek. 1997. *Statistical Methods for Speech Recognition*. MIT press.

R. Kneser and H. Ney. 1995. Improved Backing-Off for M-Gram Language Modeling. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 181–184 vol.1.

Anderson Mora-Cortes, Nikolay V Manyakov, Nikolay Chumerin, and Marc M Van Hulle. 2014. Language Model Applications to Spelling with Brain-Computer Interfaces. *Sensors*, 14(4):5967–5993.

U. Orhan, K. E. Hild, D. Erdogmus, B. Roark, B. Oken, and M. Fried-Oken. 2012. Rsvp Keyboard: An EEG Based Typing Interface. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 645–648.

Umut Orhan, Deniz Erdogmus, Brian Roark, Shalini Purwar, Kenneth E Hild, Barry Oken, Hooman Nezamfar, and Melanie Fried-Oken. 2011. Fusion with Language Models Improves Spelling Accuracy for ERP-based Brain Computer Interface spellers. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 5774–5777. IEEE.

Tom Ouyang, David Rybach, Françoise Beaufays, and Michael Riley. 2017. Mobile Keyboard Input Decoding with Finite-State Transducers. *arXiv preprint arXiv:1704.03987*.

Brian Roark, Richard Sproat, Cyril Allauzen, Michael Riley, Jeffrey Sorensen, and Terry Tai. 2012. The Opengrm Open-Source Finite-State Grammar Software Libraries. In *Proceedings of the ACL 2012 System Demonstrations*, pages 61–66, Jeju Island, Korea. Association for Computational Linguistics.

Eric W Sellers, Theresa M Vaughan, and Jonathan R Wolpaw. 2010. A Brain-Computer Interface for Long-Term Independent Home Use. *Amyotrophic Lateral Sclerosis*, 11(5):449–455.

W Speier, C Arnold, and N Pouratian. 2016. Integrating Language Models into Classifiers for BCI Communication: A Review. *Journal of Neural Engineering*, 13(3):031002.

W Speier, CW Arnold, A Deshpande, J Knall, and N Pouratian. 2015. Incorporating Advanced Language Models into the P300 Speller Using Particle Piltering. *Journal of Neural Engineering*, 12(4):046018.

William Speier, Corey Arnold, Jessica Lu, Ricky K Taira, and Nader Pouratian. 2011. Natural Language Processing with Dynamic Classification Improves P300 Speller Accuracy and Bit Rate. *Journal of Neural Engineering*, 9(1):016004.

Keith Vertanen and Per Ola Kristensson. 2011. The Imagination of Crowds: Conversational AAC Language Modeling Using Crowdsourcing and Large Data Sources. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 700–711, Stroudsburg, PA, USA. Association for Computational Linguistics.

A. Viterbi. 1967. Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.