A Framework for Decoding Event-Related Potentials from Text

Shaorong Yan
Department of Brain and Cognitive Sciences
University of Rochester
Rochester, NY 14627, USA
syan13@ur.rochester.edu

Aaron Steven White
Deparment of Linguistics
University of Rochester
Rochester, NY 14627, USA
aaron.white@rochester.edu

Abstract

We propose a novel framework for modeling event-related potentials (ERPs) collected during reading that couples pre-trained convolutional decoders with a language model. Using this framework, we compare the abilities of a variety of existing and novel sentence processing models to reconstruct ERPs. We find that modern contextual word embeddings underperform surprisal-based models but that, combined, the two outperform either on its own.

1 Introduction

Understanding the mechanisms by which comprehenders incrementally process linguistic input in real time has been a key endeavor of cognitive scientists and psycholinguists. Due to its fine time resolution, event-related potentials (ERPs) are an effective tool in probing the rapid, online cognitive processes underlying language comprehension. Traditionally, ERP research has focused on how the properties of the language input affect different ERP components (see Van Petten and Luka, 2012; Kuperberg, 2016, for reviews).[1]

While this approach has been fruitful, researchers have also long been aware of the potential drawbacks to this *component-centric* approach: a predictor's effects can be too transient to detect when averaging ERP amplitudes over a wide time window—as is typical in component-based approaches (see Hauk et al., 2006, for discussion). Different predictors can affect ERP in the same time window as an established component but have slightly different temporal (Frank and Willems, 2017) or spatial (DeLong et al., 2005) profiles. This means that the definition of a component strongly affects interpretation.

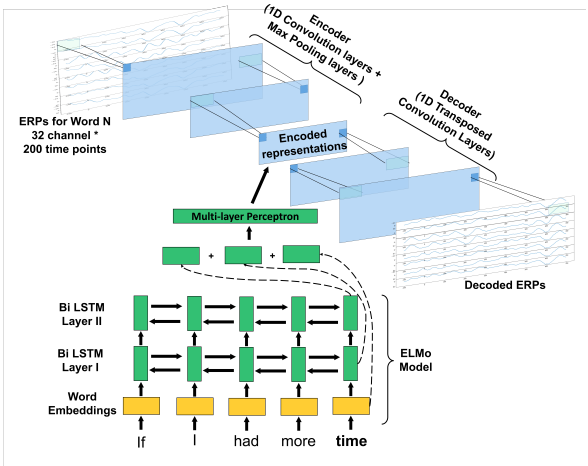

Figure 1: An instance of our framework using a bidirectional language model as the text encoder.

There are two typical approaches to resolving these issues. The first is to plot the data and use visual inspection to select an analysis plan, introducing uncontrollable researcher degrees of freedom (Gelman and Loken, 2014). Another approach is to run separate models for each time point (or even each electrode) to look for the emergence of an effect. This necessitates complex statistical tests to monitor for inflated Type I error (see, e.g., Blair and Karniski, 1993; Laszlo and Federmeier, 2014, for discussion) and to control for autocorrelation across time points (Smith and Kutas, 2015a,b).

We explore an alternative approach to the analysis of ERP data in language studies that substantially reduces such researcher degrees of freedom: directly decoding the raw electroencephalography (EEG) measurements by which ERPs are collected. Inspired by multimodal tasks like image captioning (see Hossain et al., 2019, for a review) and visual question answering (Antol et al., 2015), we propose to model EEG using standard convolutional neural networks (CNNs) pre-trained

[1] Examples of such components include the N1/P2 (Sereno et al., 1998; Dambacher et al., 2006); N250 (Grainger et al., 2006); N400 (Kutas and Hillyard, 1980; Hagoort et al., 2004; Lau et al., 2008); and P600 (Osterhout and Holcomb, 1992; Kuperberg et al., 2003; Kim and Osterhout, 2005)

under an autoencoding objective. The decoder CNN can then be decoupled from its encoder and recoupled with any language processing model, thus enabling explicit quantitative comparison of such models. We demonstrate the efficacy of this framework by using it to compare existing sentence processing models based on surprisal and/or static word embeddings with novel models based on contextual word embeddings. We find that surprisal-based models actually outperform contextual word embeddings on their own, but when combined, the two outperform either model alone.

2 Models

All of the models we present have two components: (i) a pre-trained CNN for decoding raw EEG measurements time-locked to each word in a sentence; and (ii) a language model from which features can be extracted for each word—e.g. the surprisal of that word given previous words or its contextual word embedding. An example model structure using ELMo embeddings (Peters et al., 2018) is illustrated in Figure 1.

Pre-trained convolutional decoder For all models, we use the same convolutional decoder pre-trained under an autoencoder loss. The autoencoder consists of two parts: (a) a convolutional encoder that finds a way to best compress the ERP signals; and (b) a convolutional decoder with an isomorphic architecture that reconstructs the ERP data from the compressed representation. ERPs were organized into a 2D matrix (channel $\times$ time points). For the encoder, we pass the ERPs through multiple interleaved 1D convolutional and max pooling layers with receptive fields along the time dimension, shrinking the number of latent channels at each step. Correspondingly, for the decoder part, we use an isomorphic series of 1D tranposed convolutional layers to reconstruct the ERP data. (See Appendix A for further details.)

At train time, the decoder weights are frozen, and the encoder is replaced by one of the language models described below. This entails fitting an *interface mapping*—a linear transformation for each channel produced by the encoder—from the featured extracted from the language model into the representation space output by the encoder.

Language models We consider a variety of features that can be extracted from a language model.

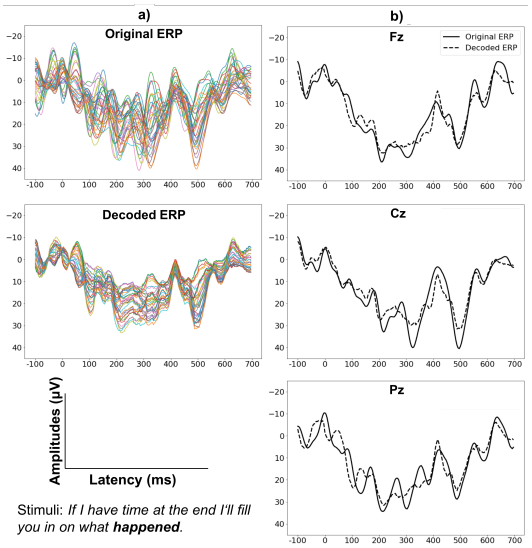

Figure 2: Original ERPs and ERPs decoded from the trained autoencoder of an example trial. a) ERPs from all 32 channels (denoted by color). b) Original (solid) and decoded ERPs (dashed) for example electrodes.

Surprisal We use the lexical surprisal $\log p(w_i \mid w_1, \ldots, w_{i-1})$ obtained from a RNN trained by Frank et al. (2015).

Semantic distance Following Frank and Willems (2017), we point-wise average the GloVe embedding (Pennington et al., 2014) of each word prior to a particular word to obtain a context embedding and then calculate the cosine distance between the context embedding and the word embedding for that word. We use the GloVe embeddings trained on Wikipedia 2014 and Gigaword 5 (6B tokens, 400K vocabulary size).

Static word embeddings We also consider the GloVe embedding dimensions as features. We do not tune the GloVe embeddings using an additional recurrent neural network (RNN), instead just passing the them through a multi-layer perceptron with one hidden layer of tanh nonlinearities. The idea here is that the GloVe-only model tells us how much the distributional-properties of a word, outside of the current context, contribute to ERPs.

Contextual word embeddings We consider contextual word embeddings generated from ELMo (Peters et al., 2018) using the `allennlp` package (Gardner et al., 2017). ELMo produces contextual word embeddings using a combination of character-level CNNs and bidirectional RNNs trained against a language modeling objective, and thus it is a useful contrast to GloVe, since it captures not only a word's distributional properties, but how they interact with the current context.

Model	R^2_{raw}	R^2_{adj}
Frequency	16.7	19.5
F + Surp	32.0	37.4
F + SemDis	30.9	36.1
F + GLoVE	29.9	35.0
F + ELMo	30.1	35.2
F + S + SD	39.9	46.6
F + S + SD + GloVe	40.3	47.1
F + S + SD + ELMo	42.3	49.5

Table 1: Proportion variance explained by each model ($\times 100$). F = frequency, S(urp) = surprisal, S(em)D(is) = semantic distance.

We take all three layers of the hidden layer output in the ELMo model and concatenate them. To ensure a fair comparison with the surprisal- and GloVe-based models, we use the same tuning procedure employed for the static word embeddings. Further, because sentences are presented incrementally in ERP experiments and because ELMo is bidirectional and thus later words in the sentence will affect the word embeddings of previous words, we do not obtain an embedding for a particular word on the basis of the entire sentence, instead using only the portion of the sentence up to and including that word to obtain its embedding.

Combined models We also consider models that combine either static or contextual word embedding features with frequency, surprisal, and semantic distance. The latter features were concatenated onto the tuned word embeddings before being passed to the interface mapping.

3 Experiments

We use the EEG recordings collected and modeled by Frank and Willems (2017). In the study, 24 subjects read sentences drawn from natural text. Sentences were presented word by word using a rapid serial visual presentation paradigm We use the ERPs of each word epoched from -100 to 700ms and time-locked to word onset from all the 32 recorded scalp channels. After artifact rejection (provided by Frank and Willems with the data), this dataset contains 41,009 training instances.

Pre-training The autoencoder is trained end-to-end with an MSE loss on the entire dataset. We employed grid search over a variety of model architectures, conducting 5-fold cross-validation for each architecture to find the one that has the best performance in reconstructing ERP data (see Appendix A for details). As shown in Figure 2, the autoencoder can reconstruct the ERP signal very well. The selected channels are illustrative of the reconstruction accuracy across all channels.

Training The interface mapping and (where applicable) word embedding tuner are trained under an MSE loss using mini-batch gradient descent (batch size = 128) with the Adam optimizer (learning rate=0.001 and default settings for beta1, beta2, and epsilon) implemented in `pytorch` (Paszke et al., 2017). Each model is trained for 200 epochs. Since we need at least one preceding word to compute contextual word embeddings, we do not include the first word of the sentence. This left ERPs for 1,618 word tokens per subject (638 word types). After excluding trials containing artifacts, a total of 37,112 training instances remain.

Development To avoid overfitting, we use early stopping and report the models with the best performance on the development set. We did a parameter search over three different weight decays: 1e-5, 1e-3, 1e-1. For each model, we chose the weight decay that produced the best mean performance on held-out data in a 5-fold cross-validation.

Baselines As a baseline we train an intercept-only model that passes a constant input (optimized to best predict the data) to the decoder. In addition, we fit a baseline model that only has word frequency as a feature. Frequency is also included as an additional feature in all models.

Metrics We report two metrics of model performance. For both, we report mean performance on held-out data in a 5-fold cross-validation (a separate). The first metric measures the total variance explained by the model under the assumption that all variance in the data could be explained:

$$R^2_{\text{raw}} = 1 - \frac{\text{MSE}_{\text{model}}}{\text{MSE}_{\text{intercept}}}$$

· The second metric accounts for the fact that our model performance is bounded by the performance of the autoencoder.

$$R^2_{\text{adj}} = 1 - \frac{\text{MSE}_{\text{model}} - \text{MSE}_{\text{autoencoder}}}{\text{MSE}_{\text{intercept}} - \text{MSE}_{\text{autoencoder}}}$$

4 Results

Table 1 shows the results. The overarching pattern is that both surprisal and semantic distance outperform both types of word embedding features, all of which outperform frequency alone. When

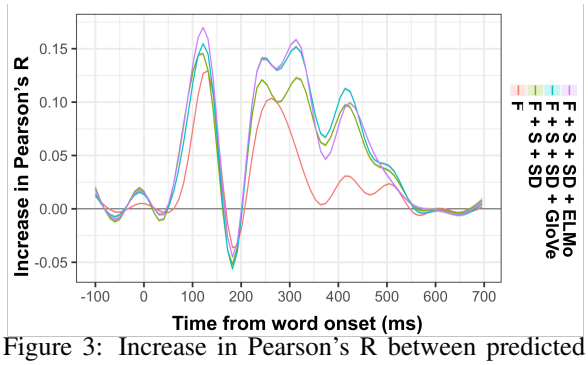

Figure 3: Increase in Pearson's R between predicted and actual ERPs. Lines show GAM smooth over time.

combined, surprisal and semantic distance outperform either alone, and further gains can be made with the addition of either static (GloVe) or contextual (ELMo) embedding features. The addition of contextual embedding features increases performance more than the addition of static word embedding features, such that there is some benefit to capturing context over and above that provided by surprisal and semantic distance.

Figure 3 plots the increase in Pearson correlation over the intercept model at each time point. There are roughly three regions where the language models outperform the baselines. The first is right after 100ms post word onset: corresponding to the N1 component, which is typically considered to reflect perceptual processing; the second is between 200 and 350ms: corresponding to the N250 component, which correlates with lexical access (Laszlo and Federmeier, 2014; Grainger et al., 2006); and the third is between 300ms and 500ms: corresponding to the N400, which is typically associated with semantic processing.[2]

In comparison to the frequency model, adding the other predictors improves the model performance the most in the N400 time window. Models with word embeddings do not differ much from the models containing only frequency, surprisal, and semantic distance, with the biggest difference being slightly before 300ms post word onset. This may indicate that processes commonly associated with the N250 may be better captured by the models containing word embeddings. If so, it is potentially interesting that these models do not differ much in the N400 time window, since one might expect at least contextual word embeddings to model semantic processing well. (See Appendix B for further analysis of model differences.)

[2]The earliest effects are less expected, since most of our models have no access to perceptual properties of the input—with the possible exception of ELMo, whose charCNN may capture orthographic regularities. These effects could reflect our models' ability to capture top-down perceptual processing (see, e.g., Penolazzi et al., 2007) or possibly systematic correlation between higher-level and perceptual features.

5 Related Work

Traditionally, ERP studies of language processing use coarse-grained predictors like cloze rates, which often lacks the precision to differentiate different neural computational models (for discussion, see Yan et al., 2017; Rabovsky et al., 2018). To overcome such limitations, a main line of attack has been to extract measures from probabilistic language models and evaluate them against ERP amplitudes (Frank et al., 2015; Brouwer et al., 2017; Rabovsky et al., 2018; Delaney-Busch et al., 2019; Fitz and Chang, 2018; Szewczyk and Wodniecka, 2018; Biemann et al., 2015).

While prior studies have also predicted ERPs from language model-based features (Broderick et al., 2018; Frank and Willems, 2017; Hale et al., 2018), they fit to aspects of the EEG signals that are unlikely to be related to language processing. Our approach threads the needle by first finding abstract structure in the ERPs with a CNN, then using that knowledge in predicting that structure from linguistic features. We are not the first to use CNNs to model EEG/ERPs (Lawhern et al., 2016; Schirrmeister et al., 2017; Seeliger et al., 2018; Acharya et al., 2018; Moon et al., 2018), but to our knowledge, no other work has yet used CNNs for modeling ERPs during reading.

6 Conclusion

We proposed a novel framework for modeling ERPs collected during reading. Using this framework, we compared the abilities of a variety of existing and novel sentence processing models to reconstruct ERPs, finding that modern contextual word embeddings underperform surprisal-based models but that, combined, the two outperform either on its own.

ERP data provides a rich testbed not only for comparing models of language processing, but potentially also for probing and improving the representations constructed by natural language processing (NLP) systems. We provided one example of how such probing might be carried out by analyzing the differences among models as a function of processing time, but this analysis only scratches the surface of what is possible using our framework, especially for understanding the more complex neural models used in NLP.

References

U Rajendra Acharya, Shu Lih Oh, Yuki Hagiwara, Jen Hong Tan, and Hojjat Adeli. 2018. Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals. *Computers in biology and medicine*, 100:270–278.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual Question Answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Chris Biemann, Steffen Remus, and Markus J Hofmann. 2015. Predicting word 'predictability' in cloze completion, electroencephalographic and eye movement data. In *Proceedings of natural language processing and cognitive science*, pages 83–93. Libreria Editrice Cafoscarina.

R Clifford Blair and Walt Karniski. 1993. An alternative method for significance testing of waveform difference potentials. *Psychophysiology*, 30(5):518–524.

Michael P Broderick, Andrew J Anderson, Giovanni M Di Liberto, Michael J Crosse, and Edmund C Lalor. 2018. Electrophysiological correlates of semantic dissimilarity reflect the comprehension of natural, narrative speech. *Current Biology*, 28(5):803–809.

Harm Brouwer, Matthew W Crocker, Noortje J Venhuizen, and John CJ Hoeks. 2017. A neurocomputational model of the N400 and the P600 in language processing. *Cognitive Science*, 41:1318–1352.

Michael Dambacher, Reinhold Kliegl, Markus Hofmann, and Arthur M Jacobs. 2006. Frequency and predictability effects on event-related potentials during reading. *Brain Research*, 1084(1):89–103.

Nathaniel Delaney-Busch, Emily Morgan, Ellen Lau, and Gina R Kuperberg. 2019. Neural evidence for bayesian trial-by-trial adaptation on the N400 during semantic priming. *Cognition*, 187:10–20.

Katherine A DeLong, Thomas P Urbach, and Marta Kutas. 2005. Probabilistic word pre-activation during language comprehension inferred from electrical brain activity. *Nature Neuroscience*, 8(8):1117–1121.

Hartmut Fitz and Franklin Chang. 2018. Sentence-level erp effects as error propagation: A neurocomputational model. *PsyArXiv*.

Stefan L Frank, Leun J Otten, Giulia Galli, and Gabriella Vigliocco. 2015. The ERP response to the amount of information conveyed by words in sentences. *Brain and Language*, 140:1–11.

Stefan L Frank and Roel M Willems. 2017. Word predictability and semantic similarity show distinct patterns of brain activity during language comprehension. *Language, Cognition and Neuroscience*, 32(9):1192–1203.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. AllenNLP: A Deep Semantic Natural Language Processing Platform.

Andrew Gelman and Eric Loken. 2014. The statistical crisis in science. *The best writing on mathematics*, 102(6):460–465.

Jonathan Grainger, Kristi Kiyonaga, and Phillip J Holcomb. 2006. The time course of orthographic and phonological code activation. *Psychological Science*, 17(12):1021–1026.

Peter Hagoort, Lea Hald, Marcel Bastiaansen, and Karl Magnus Petersson. 2004. Integration of word meaning and world knowledge in language comprehension. *Science*, 304(5669):438–441.

John Hale, Chris Dyer, Adhiguna Kuncoro, and Jonathan Brennan. 2018. Finding syntax in human encephalography with beam search. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2727–2736. Association for Computational Linguistics.

Olaf Hauk, Matthew H Davis, M Ford, Friedemann Pulvermüller, and William D Marslen-Wilson. 2006. The time course of visual word recognition as revealed by linear regression analysis of ERP data. *Neuroimage*, 30(4):1383–1400.

MD. Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. 2019. A comprehensive survey of Deep Learning for Image Captioning. *ACM Comput. Surv.*, 51(6):118:1–118:36.

Albert Kim and Lee Osterhout. 2005. The independence of combinatory semantic processing: Evidence from event-related potentials. *Journal of Memory and Language*, 52(2):205–225.

Gina R Kuperberg. 2016. Separate streams or probabilistic inference? what the N400 can tell us about the comprehension of events. *Language, Cognition and Neuroscience*, 31(5):602–616.

Gina R Kuperberg, Tatiana Sitnikova, David Caplan, and Phillip J Holcomb. 2003. Electrophysiological distinctions in processing conceptual relationships within simple sentences. *Cognitive Brain Research*, 17(1):117–129.

Marta Kutas and Steven A Hillyard. 1980. Reading senseless sentences: Brain potentials reflect semantic incongruity. *Science*, 207(4427):203–205.

Sarah Laszlo and Kara D Federmeier. 2014. Never seem to find the time: evaluating the physiological time course of visual word recognition with regression analysis of single-item event-related potentials. *Language, Cognition and Neuroscience*, 29(5):642–661.

Ellen F Lau, Colin Phillips, and David Poeppel. 2008. A cortical network for semantics:(de) constructing the N400. *Nature Reviews Neuroscience*, 9(12):920.

Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. 2016. EEGnet: A compact convolutional network for EEG-based brain-computer interfaces. *arXiv preprint arXiv:1611.08024*.

Seong-Eun Moon, Soobeom Jang, and Jong-Seok Lee. 2018. Convolutional neural network approach for EEG-based emotion recognition using brain connectivity and its spatial information. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2556–2560. IEEE.

Lee Osterhout and Phillip J Holcomb. 1992. Event-related brain potentials elicited by syntactic anomaly. *Journal of Memory and Language*, 31(6):785–806.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Barbara Penolazzi, Olaf Hauk, and Friedemann Pulvermüller. 2007. Early semantic context integration and lexical access as revealed by event-related brain potentials. *Biological Psychology*, 74(3):374–388.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 2227–2237.

Milena Rabovsky, Steven S Hansen, and James L McClelland. 2018. Modelling the N400 brain potential as change in a probabilistic representation of meaning. *Nature Human Behaviour*, 2(9):693.

Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball. 2017. Deep learning with convolutional neural networks for EEG decoding and visualization. *Human Brain Mapping*, 38(11):5391–5420.

Katja Seeliger, Matthias Fritsche, Umut Güçlü, Sanne Schoenmakers, Jan-Mathijs Schoffelen, Sander Bosch, and Marcel van Gerven. 2018. Convolutional neural network-based encoding and decoding of visual object recognition in space and time. *NeuroImage*, 180:253–266.

Sara C Sereno, Keith Rayner, and Michael I Posner. 1998. Establishing a time-line of word recognition: evidence from eye movements and event-related potentials. *Neuroreport*, 9(10):2195–2200.

Nathaniel J Smith and Marta Kutas. 2015a. Regression-based estimation of ERP waveforms: I. the rERP framework. *Psychophysiology*, 52(2):157–168.

Nathaniel J Smith and Marta Kutas. 2015b. Regression-based estimation of ERP waveforms: Ii. nonlinear effects, overlap correction, and practical considerations. *Psychophysiology*, 52(2):169–181.

Jakub M Szewczyk and Zofia Wodniecka. 2018. Mechanisms of prediction updating - preprint. *PsyArXiv*.

Cyma Van Petten and Barbara J Luka. 2012. Prediction during language comprehension: Benefits, costs, and ERP components. *International Journal of Psychophysiology*, 83(2):176–190.

Shaorong Yan, Gina R Kuperberg, and T Florian Jaeger. 2017. Prediction (or not) during language processing. a commentary on Nieuwland et al.(2017) and Delong et al.(2005). *bioRxiv*, page 143750.

A Autoencoder selection

We compared the performance of two model structures with different sized of the encoded state. The first class has 5 latent channels and 9 time steps. Given the sampling rate and size of the input ($250Hz$, 200 time steps), this thus roughly corresponds to filtering the EEG data with alpha band frequency ($10Hz$, hence, *alpha model*). Another class has 10 latent channels and 20 time steps, this lies with the range of beta band activity ($25Hz$, hence, *beta model*).

We also examined whether including subject-specific intercept would improve the model performance. To do this, we create an intercept for each electrode and each subject. During training, we first transformed the model by subtracting the subject-specific intercepts from the raw ERPs. For the decoder, we added back subject-specific intercepts to the decoded ERPs before evaluated them against the raw EEGs. The subject-specific intercepts were also adjusted together with the other parameters of the autoencoder.

The performance of each set of models can be found in Table 2. As expected, beta models overall perform better than alpha models, since it likely captures alpha band activities. Adding subject-specific intercept, on the other hand, did not greatly improve the model performance. For our combined models, we thus selected the beta model without subject-specific intercept.

Model	Without Intercept	With Intercept
alpha	$49.9(53.2)$	$49.7(53.3)$
beta	$33.5(68.6)$	$32.7(69.2)$

Table 2: Mean MSE and R^2_{raw} (in parentheses) of each model.

B Additional analysis of model differences

We also examined the performance of each model on different parts-of-speech. We calculated the Pearson's R between the predicted and actual ERPs for each word of each model and used linear mixed-effects model to examine the influence on model fit with the inclusion of different information. If a model included a specific type of information, the corresponding predictor is coded as 1, otherwise it was coded as -1. For example, the surprisal model was trained with surprisal but not semantic distance, so the surprisal predictor is 1 for this model and the semantic distance predictor is -1.

We first examined whether each type of information has different influence on content words vs. function words. The results are presented in Table 3. Function words were coded as -1 and content words were coded as 1. Overall models had better performance for content words than for function words ($\hat{\beta} = 0.003$, t $= 2.01, p < 0.05$). Including each type of information also significantly increased model fit (ts $> 10.1, p < 0.01$). There was a significant interaction between frequency and word type ($\hat{\beta} = -0.001$, t $= -2.43, p < 0.02$) such that including frequency increased model performance for function words more than for content words. There was also a marginally significant interaction between ELMo and word type ($\hat{\beta} = 0.0007$, t $= -1.85, p < 0.064$) such that including ELMo embeddings increased model performance for content words more than for function words.

Predictor	$\hat{\beta}$	t	
Intercept	-0.0013	-0.225	
Word Type (Content)	0.003	2.01	*
Frequency	0.011	21.2	**
Surprisal	0.005	13.0	**
Semantic Distance	0.004	11.60	**
GloVe Embeddings	0.004	10.3	**
ELMo Embeddings	0.004	10.1	**
Freq : Word Type	-0.001	-2.43	*
Surp : Word Type	0.0001	0.24	
SemDis : Word Type	-0.0003	-0.70	
GloVe : Word Type	0.0002	0.55	
ELMo : Word Type	0.0007	-1.85	$^+$

Table 3: Model estimates and t statistics from mixed-effects model. $^{**}: p < 0.01$; $^*: p < 0.05$; $^+: p < 0.1$

We also examined the interaction between each type of information and each part-of-speech. Overall the models had worse performance for particles ($\hat{\beta} = -0.017$, t $= -3.37, p < 0.01$), nouns ($\hat{\beta} = -0.007$, t $= -1.95, p < 0.051$) and pronouns ($\hat{\beta} = -0.012$, t $= -1.76, p < 0.08$). Including each type of information increased overall model fit (ts $> 6.05, p < 0.01$). While including frequency increased overall model fit, it increased the model fit for verbs less ($\hat{\beta} = -0.003$, t $= -2.04, p < 0.05$). No other effects reached significance.