PIHKers at CMCL 2021 Shared Task: Cosine Similarity and Surprisal to Predict Human Reading Patterns.

Lavinia Salicchi
The Hong Kong Polytechnic University
lavinia.salicchi@connect.polyu.hk

Alessandro Lenci
Università di Pisa
alessandro.lenci@unipi.it

Abstract

Eye-tracking psycholinguistic studies have revealed that context-word semantic coherence and predictability influence language processing. In this paper we show our approach to predict eye-tracking features from the ZuCo dataset for the shared task of the Cognitive Modeling and Computational Linguistics (CMCL2021) workshop. Using both cosine similarity and surprisal within a regression model, we significantly improved the baseline Mean Absolute Error computed among five eye-tracking features.

1 Introduction

The shared task proposed by the organizers of the Cognitive Modeling and Computational Linguistics workshop (Hollenstein et al., 2021) requires participant to create systems capable of predicting eye-tracking data from the ZuCo dataset (Hollenstein et al., 2018). Creating systems to efficiently predict biometrical data may be useful to make prediction about linguistic materials for which we have few or none experimental data, and to make new hypothesis about the internal dynamics of cognitive processes.

The approach we propose relies mainly on two factors that have been proved to influence language comprehension: i.) the **semantic coherence** of a word with the previous ones (Ehrlich and Rayner, 1981) and ii.) its **predictability** from previous context (Kliegl et al., 2004). We model the first factor with the *cosine similarity* (Mitchell et al., 2010; Pynte et al., 2008) between the distributional vectors, representing the context and the target word, produced by different Distributional Semantic Models (DSM) (Lenci, 2018). We compared 10 state-of-the-art word embedding models, and two different approaches to compute the context vector. We model the predictability of a word within the context with the word-by-word *surprisal* computed with 3 of the above mentioned models (Hale, 2001; Levy, 2008). Finally, cosine similarity and surprisal are combined in different regression models to predict eye tracking data.

2 Related Works

Different word embedding models (GloVe, Word2Vec, WordNet2Vec, FastText, ELMo, BERT) have been evaluated in the framework proposed by Hollenstein et al. (2019). The evaluation is based on the model capability to reflect semantic representations in the human mind, using cognitive data in different datasets for eye-tracking, EEG, and fMRI. Word embedding models are used to train neural networks on a regression task. The results of their analyses show that BERT, ELMo, and FastText have the best prediction performances.

Regression models with different combinations of cosine similarity and surprisal, to predict (and further study the cognitive dynamics beneath) eye movements have been created by Frank (2017), who claims that, since word embeddings are based on co-occurrences, semantic distance may actually represent word predictability, rather than semantic relatedness, and that previous findings showing correlations between reading times and semantic distance were actually due to a confound between these two concepts. In his work, he uses linear regression models testing different surprisal measures, and excluding it. The results show that when surprisal is factored out, the effects of semantic similarity on reading times disappear, proving thus the existence of an interplay between the two elements.

3 Experimental Setting

3.1 Datasets

The shared task materials come from ZuCo (Hollenstein et al., 2018), that includes EEG and eye-tracking data, collected on 12 English speakers reading natural texts. The data collection has been

Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, pages 102–107
Online Event, June 10, 2021. ©2021 Association for Computational Linguistics

done in three different settings: two normal reading tasks and one task-specific reading session. The original dataset comprises $1,107$ sentences, and for the shared task 800 sentences ($15,736$ words) have been used for the training data, while the test set included about 200 sentences ($3,554$ words). Since the shared task focuses on eye-tracking features, only this latter data were available. The training dataset structure includes sentence number, word-within-sentence number, word, number of fixations (nFix), first fixation duration (FFD), total reading time (TRT), go-past time (GPT), fixation proportion (fixProp). The first three elements were part of the test set too.

Our approach includes a preliminary step of feature selection. For this purpose we also used GECO (Cop et al., 2017) and Provo (Luke and Christianson, 2018), two eye-tracking corpora containing long, complete, and coherent texts. **GECO** is a monolingual and bilingual (English and Dutch) corpus composed of the entire Agatha Christie's novel *The Mysterious Affair at Styles*. GECO contains eye-tracking data of 33 subjects (19 of them bilingual, 14 English monolingual) reading the full novel text, presented paragraph-by-paragraph on a screen. GECO is composed of $54,364$ tokens. **Provo** contains 55 short English texts about various topics, for a total of $2,689$ tokens, and a vocabulary of $1,197$ words. These texts were read by 85 subjects and their eye-tracking measures were collected in an available on-line dataset. Similarly to ZuCo, GECO and Provo data are recorded during naturalistic reading on everyday life materials. For every word in GECO and Provo, we extracted its mean total reading time, mean first fixation duration, and mean number of fixations, by averaging over the subjects.

3.2 Word Embeddings

Table 1 shows the embeddings types used in our experiments, consisting of 6 non-contextualized DSMs and 4 contextualized DSMs. The former include predict models (**SGNS** and **FastText**) (Mikolov et al., 2013; Levy and Goldberg, 2014; Bojanowski et al., 2017) and count models (**SVD** and **GloVe**) (Bullinaria and Levy, 2012; Pennington et al., 2014). Four DSMs are window-based and two are syntax-based (**synt**). Embeddings have 300 dimensions and were trained on the same corpus of about 3.9 billion tokens, which is a concatenation of ukWaC and a 2018 dump of Wikipedia.

Pre-trained contextualized embeddings include the 512-dimensional vectors produced by the three layers of the **ELMo** bidirectional LSTM architecture (Peters et al., 2018), the $1,024$-dimensional vectors in the 24 layers of **BERT-Large** Transformers (BERT-Large, Cased) (Devlin et al., 2019), the $1,600$-dimensional vectors of **GPT2-xl** (Radford et al.), and the 200-dimensional vectors produced by the **Neural Complexity** model (van Schijndel and Linzen, 2018).

3.3 Method

To predict eye tracking data we tested different regression models and several features combinations.

Feature Selection. To select the features to be used, for each word embedding model and language model we carried out a preliminary investigation computing Spearman's correlation between eye tracking features, and respectively surprisal and cosine similarity: The features with the highest correlation with biometrical data have been selected for being used in the regression model.

For each target word w in GECO, Provo and ZuCo, we measure the **cosine similarity** between the embedding of w and the embedding of the context c composed of the previous words in the same sentence. We then compute the Spearman correlation between the cosine and the eye-tracking data for w. We test two different ways of computing the context embedding:

Additive model (for every embedding type): The context vector is the sum of all its word embeddings. Because of the bidirectional nature of BERT, the input to this model needed a special pre-processing. In order to prevent that the vectors representing words within the context were computed using the target word itself, we passed to BERT a list of sub-sentences, each of which were composed of context words only. So given the sentence *The dog chases the cat*:

S[0] = ["The"]
S[1] = ["The dog"]
S[2] = ["The dog chases"]
S[3] = ["The dog chases the"]
S[4] = ["The dog chases the cat"]

Starting from the second sub-sentence, the cosine similarity is computed between the last word vector and the sum of words vectors belonging to the previous sub-sentence (list element). Therefore, to compute the cosine similarity between *cat* and the previous context, we select *cat* from S[4] and

Model	Hyperparameters
Non-contextualized DSMs	
SVD.w2	count DSM with 345K window-selected context words, window of width 2, reduced with SVD
SVD.synt	count DSM with 345K syntactically typed context words reduced with SVD
GloVe	count DSM with context window of width 2, reduced with log-bilinear regression
SGNS.w2	Skip-gram with negative sampling, context window of width 2, 15 negative examples
SGNS.synt	Skip-gram with negative sampling, syntactically-typed context words, 15 negative examples
FastText	Skip-gram with subword information, context window of width 2, 15 negative examples
Contextualized DSMs	
ELMo	Pretrained ELMo embeddings on the 1 Billion Word Benchmark
BERT	Pretrained BERT-Large embeddings on the concatenation of the Books corpus and Wikipedia
GPT2-xl	Pretrained GPT2-xl embeddings on WebText
Neural Complexity	Pretrained Neural Complexity embeddings on Wikipedia

Table 1: List of the embedding models used for the study, together with their hyperparameter settings.

$The + dog + chases + the$ from S[3].

CLS: The context vector is the embedding produced by BERT for the special token [CLS]. As for the additive model, BERT was fed with sub-sentences, and for each target word the CLS-context-vector was the one computed at the previous list element. So, looking at the previous example, for *cat* as target word, we will use the CLS vector representing all the S[3] elements.

Given the positive effect of semantic coherence on language processing, we expect that the eye-tracking data for w have a negative correlation with its cosine similarity with c: *The higher the cosine, the lower the reading time of w measured by eye-tracking*.

We then used BERT, GPT2-xl and Neural Complexity to compute word-by-word surprisal. As for the cosine similarity, for BERT the input sentences were organized in sub-sentences, and the last token, the target word, was replaced with the special tag [MASK]. Finally, we compute the Spearman correlation between the **surprisal** of w, and the eye-tracking data for the target word. Differently from the cosine, we expect the surprisal to be positively correlated with the word reading time: *The less predictable a word, the slower its processing*.

The comparison has been done between 60 possible features: 6 values of cosine similarity between non-contextualized vectors, 51 values of cosine similarity between contextualized vectors (48 from 24 layers of BERT in two different ways to compute the context vector, and 3 from ELMo, GPT2-xl and Neural Complexity), 3 values of surprisal from BERT, GPT2-xl, Neural Complexity. Based on the correlation values, we selected one cosine similarity feature and one surprisal feature, that have been combined with two variables that are well-known in the cognitive neuroscience literature for influencing eye movements: word length and word frequency, the last one computed on Wikipedia[1].

Regression Model Selection. Taking into account the Spearman's correlations, we selected one word embedding model for cosine similarity and one Language Model for surprisal. Then, different kind of regression models from Scikit-learn have been compared. More precisely, *PLS Regression, Multi-layer Perceptron Regressor, Random Forest Regressor, Linear Regression, Ridge Regression, Bayesian ridge regression, Epsilon-Support Vector Regression, Linear regression with combined L1 and L2 priors as Regularizer, Gradient Boosting Regressor*. The **metric** used to evaluate different models is the Mean Absolute Error on ZuCo's eye tracking features prediction. Once the model and the features have been selected, the **comparison** between 3 different regression settings has been done: i) surprisal only; ii) cosine similarity only; iii) surprisal + cosine similarity.

For the regression model selection, we used 2/3 of the ZuCo training set to train the model, and 1/3 for validation purposes. Once we found the best (i.e. lower MAE among eye tracking data) combination of features and regression model, the prediction on test data has been done.

4 Results and Discussion

Spearman's correlations between eye tracking features and cosine similarity showed that best performances are reached by vectors produced by BERT layer 22 CLS context (mean correlation over eye tracking features on the three datasets: -0.62), while best correlations between eye tracking data and surprisal are reached by GPT2-xl (mean correlation over eye tracking features on the three datasets: 0.40). These results led us to select as

[1] Using https://github.com/IlyaSemenov/wikipedia-word-frequency

Feature	Model	Regression model	MAE
FFD	BERTcos_GPTsurpr	GBR	0.69
FFD	GPTcos_GPTsurpr	GBR	0.69
FFD	BERTcos_GPTsurpr	RF	0.74
FFD	BASELINE	RF	0.77
fixprop	BERTcos_GPTsurpr	GBR	11.64
fixprop	GPTcos_GPTsurpr	GBR	11.78
fixprop	GPTcos_GPTsurpr	RF	12.33
fixprop	BASELINE	RF	12.75
GPT	BERTcos_GPTsurpr	GBR	2.96
GPT	GPTcos_GPTsurpr	GBR	2.978
GPT	BERTcos_GPTsurpr	LR	3.08
GPT	BASELINE	BRR	3.09
nFix	BERTcos_GPTsurpr	GBR	4.21
nFix	GPTcos_GPTsurpr	GBR	4.37
nFix	BERTcos_GPTsurpr	LR	4.49
nFix	BASELINE	LR	4.67
TRT	BERTcos_GPTsurpr	GBR	1.64
TRT	GPTcos_GPTsurpr	GBR	1.67
TRT	BERTcos_GPTsurpr	RF	1.76
TRT	BASELINE	RF	1.84

Table 2: Best three MAEs for each eye-tracking feature + baseline.

features for regression model: cosine similarity between vectors computed by BERT 22 CLS and surprisal computed by GPT2-xl. We also tested the cosine similarity between vectors computed by GPT2-xl, to have a comparison with a regression model with features produced by the same model. While performing regression model selection comparing 9 models from Scikit-learn, we also tried different combinations of features.

Table 2 shows the best 3 combinations of features and models, compared with the baseline created taking into account word frequency and word lenght only. The lowest MAEs for each eye-tracking feature were reached by a Gradient Boosting Regressor (GBR) using both the cosine similarity between vectors produced by BERT and the surprisal computed by GPT2-xl. The average MAE using the GBR model with BERT cosine and GPT2-xl surprisal was 4.22 (mean improvement compared with the baseline $= 0.54$), with one feature, *fixProp*, producing a MAE value significantly higher than the other eye tracking features. Since fixProp is "*the proportion of participants that fixated the current word*" (i.e., the probability of the word of being fixed), we hypothesized that the combination of phenomena influencing the likelihood of fixating a word could be captured by the other 4 eye tracking features, making them in turn good predictors of fixProp.

Therefore, we tested again the 9 regression models with Scikit-learn, this time using nFix, FFD, TRT, GPT, word lenght and word frequency as features, in every possible permutation (one per time, pairs of features, etc.). A lower MAE on fixProp on training data has been obtained using a *Random Forest* method with nFix, TRT, and GPT, reaching a MAE of 3.15.

The improvements of the final model over the baseline suggest that the information conveyed by the cosine similarity and the surprisal contributes in modeling the cognitive processing beneath reading. Our results are consistent with Pynte et al. (2008) and Mitchell et al. (2010) findings about the relation between cosine similarity and eye movements data, as well as with Hale (2001) and Levy (2008), who found surprisal to be useful in predicting reading times.
Anyway, our model performance shows that taking into account *both* the computational measures benefits the modeling. Even if Frank (2017) rises an interesting issue about the interplay between the information included in word embeddings and the one provided by the suprisal computed by language models, our results keep us from fully agree with his observations: since the joined model performed better that the ones taking into account only cosine similarity or only surprisal, it is obvious that the two measures convey exclusive and useful information, even if it is more than plausible that they share some kind of information to some extent.

In summary, we used a two-step approach: i.) the final model to predict nFix, FFD, GPT, and TRT in test data was a Gradient Boosting Regressor having as features the cosine similarity between the CLS vector (BERT) and the target word embedding, GPT2-xl surprisal, word length and word frequency; ii.) the predicted values of nFix, GPT, and TRT were used in a Random Forest to predict fixProp.

The shared task final results over the test data, revealed that our model had an average MAE of 4.3877 over all eye tracking features (the baseline was 7.3699, while the best model reached a MAE of 3.8134).

5 Conclusions

In this paper we described the system we proposed in the CMCL2021 "Shared Task: Predicting human reading patterns". We were required to create a model capable of predicting number of fixations, first fixation duration, total reading time, go-past time, and fixation proportion of each word in the ZuCo dataset. We proposed a regression model using word length and word frequency, combined with two elements that are proved to influence reading processing: the semantic coherence and the predictability of a word within the context. To compute these last two regression features we used the cosine similarity between the vector representing the context and the word embedding of the target word, and the surprisal computed by Language Models, respectively. We selected the models to produce the vectors and to compute the surprisal calculating the Spearman correlation between the cosine similarity and the eye tracking data, and between the surprisal and the same data. We then used the best cosine similarity and surprisal within a regression model, selected among 9 possible models. Our results outperformed the baseline, with a average MAE among eye tracking features just 0.5743 higher than the best model in the competition.

Our model may be improved exploring new types of regressors and word embeddings, and including new textual features such as sentence length and information regarding words immediately preceding the target ones.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

John A Bullinaria and Joseph P Levy. 2012. Extracting Semantic Representations from Word Co-Occurrence Statistics: Stop-Lists, Stemming, and SVD. *Behavior Research Methods*, 44(3):890–907.

Uschi Cop, Nicolas Dirix, Denis Drieghe, and Wouter Duyck. 2017. Presenting GECO: An Eye-Tracking Corpus of Monolingual and Bilingual Sentence Reading. *Behavior Reseach Methods*, 49(2):602–615.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL*.

Susan E. Ehrlich and Keith Rayner. 1981. Contextual Effects on Word Perception and Eye Movements During Reading. *Journal of Verbal Learning and Verbal Behavior*, 20:641–65.

Stefan L Frank. 2017. Word Embedding Distance Does not Predict Word Reading Time. In *Proceedings of CogSci*, pages 385–390.

John Hale. 2001. A Probabilistic Earley Parser as a Psycholinguistic Model. In *Proceedings of NAACL*.

Nora Hollenstein, Emmanuele Chersoni, Cassandra Jacobs, Yohei Oseki, Laurent Prévot, and Enrico Santus. 2021. Cmcl 2021 shared task on eye-tracking prediction. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*.

Nora Hollenstein, Antonio de la Torre, Nicolas Langer, and Ce Zhang. 2019. CogniVal: A Framework for Cognitive Word Embedding Evaluation. In *Proceedings of CONLL*.

Nora Hollenstein, Jonathan Rotsztejn, Marius Troendle, Andreas Pedroni, Ce Zhang, and Nicolas Langer. 2018. Zuco, a simultaneous eeg and eye-tracking resource for natural sentence reading. *Scientific Data*, 5.

Reinhold Kliegl, Ellen Grabner, Martin Rolfs, and Ralf Engbert. 2004. Length, frequency, and predictability effects of words on eye movements in reading. *European Journal of Cognitive Psychology - EUR J COGN PSYCHOL*, 16:262–284.

Alessandro Lenci. 2018. Distributional Models of Word Meaning. *Annual Review of Linguistics*, 4:151–171.

Omer Levy and Yoav Goldberg. 2014. Dependency-Based Word Embeddings. In *Proceedings of ACL*.

Roger Levy. 2008. Expectation-based Syntactic Comprehension. *Cognition*, 106(3):1126–1177.

Steven G Luke and Kiel Christianson. 2018. The Provo Corpus: A Large Eye-tracking Corpus with Predictability Norms. *Behavior Research Methods*, 50(2):826–833.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*.

Jeff Mitchell, Mirella Lapata, Vera Demberg, and Frank Keller. 2010. Syntactic and Semantic Factors in Processing Difficulty: An Integrated Measure. In *Proceedings of ACL*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of EMNLP*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of NAACL*.

Joel Pynte, Boris New, and Alan Kennedy. 2008. Online Contextual Influences During Reading Normal Text: A Multiple-Regression Analysis. *Vision research*, 48(21):2172–2183.

A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. In *Open-AI Blog*.

Marten van Schijndel and Tal Linzen. 2018. A Neural Model of Adaptation in Reading. In *Proceedings of EMNLP*.