LAST at CMCL 2021 Shared Task: Predicting Gaze Data During Reading with a Gradient Boosting Decision Tree Approach

Yves Bestgen
Laboratoire d'analyse statistique des textes - LAST
Institut de recherche en sciences psychologiques
Université catholique de Louvain
Place Cardinal Mercier, 10 1348 Louvain-la-Neuve, Belgium
yves.bestgen@uclouvain.be

Abstract

A LightGBM model fed with target word lexical characteristics and features obtained from word frequency lists, psychometric data and bigram association measures has been optimized for the 2021 CMCL Shared Task on Eye-Tracking Data Prediction. It obtained the best performance of all teams on two of the five eye-tracking measures to predict, allowing it to rank first on the official challenge criterion and to outperform all deep-learning based systems participating in the challenge.

1 Introduction

This paper describes the system proposed by the Laboratoire d'analyse statistique des textes (LAST) for the Cognitive Modeling and Computational Linguistics (CMCL) Shared Task on Eye-Tracking Data Prediction. This task is receiving more and more attention due to its importance in modeling human language understanding and improving NLP technology (Hollenstein et al., 2019; Mishra and Bhattacharyya, 2018).

As one of the objectives of the organizers is to "compare the capabilities of machine learning approaches to model and analyze human patterns of reading" (https://cmclorg.github.io/shared_task), I have chosen to adopt a generic point of view with the main objective of determining what level of performance can achieve a system derived from the one I developed to predict the lexical complexity of words and polylexical expressions (Shardlow et al., 2021). That system was made up of a gradient boosting decision tree prediction model fed with features obtained from word frequency lists, psychometric data, lexical norms and bigram association measures. If there is no doubt that predicting lexical complexity is a different problem, one can think that the features useful for it also play a role in predicting eye movement during reading.

The next section summarizes the main characteristics of the challenge. Then the developed system is described in detail. Finally, the results in the challenge are reported along with an analysis performed to get a better idea of the factors that affect the system performance.

2 Data and Task

The eye-tracking data for this shared task were extracted from the Zurich Cognitive Language Processing Corpus (ZuCo 1.0 and ZuCo 2.0, Hollenstein et al., 2018, 2020). It contains gaze data for 991 sentences read by 18 participants during a normal reading session. The learning set consisted in 800 sentences and the test set in 191 sentences.

The task was to predict five eye-tracking features, averaged across all participants and scaled in the range between 0 and 100, for each word of a series of sentences: (1) the total number of fixations (nFix), (2) the duration of the first fixation (FFD), (3) the sum of all fixation durations, including regressions (TRT), (4) the sum of the duration of all fixations prior to progressing to the right, including regressions to previous words (GPT), and (5) the proportion of participants that fixated the word (fixProp). These dependent variables (DVs) are described in detail in Hollenstein et al. (2021). The submissions were evaluated using the mean absolute error (MAE) metric and the systems were ranked according to the average MAE across all five DVs, the lowest being the best.

As the DVs are of different natures (number, proportion and duration), their mean and variance are very different. The mean of fixProp is 21 times greater than that of FFD and its variance 335 times. Furthermore, while nFix and fixProp were scaled independently, FFD, GPT and TRT were scaled together. For that reason, the mean and dispersion of these three measures are quite different: FFD must necessarily be less than or equal to TRT and GPT[1].

[1]The relation between TRT and GPT is not obvious to me

Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, pages 90–96

These two factors strongly affect the importance of the different DVs in the final ranking.

3 System

3.1 Procedure to Build the Models

The regression models were built by the 2.2.1 version of the LightGBM software (Ke et al., 2017), a well-known implementation of the gradient boosting decision tree approach. This type of model has the advantage of not requiring feature preprocessing, such as a logarithmic transformation, since it is insensitive to monotonic transformations, and of including many parameters allowing a very efficient overfit control. It also has the advantage of being able to directly optimize the MAE.

Sentence preprocessing and feature extraction as well as the post-processing of the LightGBM predictions were performed using custom SAS programs running in SAS University (still freely available for research at https://www.sas.com/en_us/software/university-edition.html). Sentences were first lemmatized by the TreeTagger (Schmid, 1994) to get the lemma and POS-tag of each word. Special care was necessary to match the TreeTagger tokenization with the Zuco original one. Punctuation marks and other similar symbols (e.g., "(" or "$") were simply disregarded as they were always bound to a word in the tokens to predict. The attribution to the words of the values on the different lists was carried out in two stages: on the basis of the spelling form when it is found in the list or of the lemma if this is not the case.

The features used in the final models as well as the LightGBM parameters were optimized by a 5-fold cross validation procedure, using the sentence and not the token as the sampling unit. The number of boosting iterations was set by using the LightGBM early stopping procedure which stops training when the MAE on the validation fold does not improve in the last 200 rounds. The predicted values which were outside the [0, 100] interval were brought back in this one, which makes it possible to improve the MAE very slightly.

3.2 Features

To predict the five DVs, five different models were trained. The only differences between them were in the LightGBM parameters. There were thus all based on exactly the same features, which are described below.

Target Word Length. The length in characters of the preceding word, the target word and the following one.

Target Word Position. The position of the word in the sentence encoded in two ways: the rank of the word going from 1 to the sentence total number of words and the ratio between the rank of the word and the total number of words.

Target Word POS-tag and Lemma. The POS-tag and lemma for the target word and the preceding one.

Corpus Frequency Features. Frequencies in corpora of words were either calculated from a corpus or extracted from lists provided by other researchers. The following seven features have been used:

- The (unlemmatized) word frequencies in the British National Corpus (BNC, http://www.natcorp.ox.ac.uk/).
- The Facebook frequency norms for American English and British English in Herdagdelen and Marelli (2017).
- The Rovereto Twitter Corpus frequency norms (Herdagdelen and Marelli, 2017).
- The USENET Orthographic Frequencies from Shaoul and Chris (2006).
- The Hyperspace Analogue to Language (HAL) frequency norms provided by (Balota et al., 2007) for more that 40,000 words.
- The frequency word list derived from Google's ngram corpora available at https://github.com/hackerb9/gwordlist.

Features from Lexical Norms. The lexical norms of Age of Acquisition and Familiarity were taken from the Glasgow Norms which contain judges' assessment of 5,553 English words (Scott et al., 2019).

Lexical Characteristics and Behavioral Measures from ELP. Twenty-three indices were extracted from the English Lexicon Project (ELP, Balota et al., 2007; Yarkoni et al., 2008), a database

since one can be larger or smaller than the other in a significant number of cases.

that contains, for more than 40,000 words, reaction time and accuracy during lexical decision and naming tasks, made by many participants, as well as lexical characteristics (https://elexicon.wustl.edu/). Eight indices come from the behavioral measures, four for each task: average response latencies (raw and standardized), standard deviations, and accuracies. Fourteen indices come from the "Orthographic, Phonological, Phonographic, and Levenshtein Neighborhood Metrics" section of the dataset. These are all the metrics provided except Freq_Greater, Freq_G_Mean, Freq_Less, Freq_L_Mean, and Freq_Rel. These are variables whose initial analyzes showed that they were redundant with those selected. The last feature is the average bigram count of a word.

Bigram Association Measures. These features indicate the degree of association between the target word and the one that precedes it according to a series of indices calculated on the basis of the frequency in a reference corpus (i.e., the BNC) of the bigram and that of the two words that compose it, using the following association measures (AMs): pointwise mutual information and t-score (Church and Hanks, 1990), z-score (Berry-Rogghe, 1973), log-likelihood Chi-square test (Dunning, 1993), simple-ll (Evert, 2009), Dice coefficient (Kilgarriff et al., 2014) and the two delta-p (Kyle et al., 2018). Most of the formulas to compute these AMs are also provided in Evert (2009) and in Pecina (2010). As these features mix together the assets of both collocations (by using association scores) and ngrams (by using contiguous pairs of words), Bestgen and Granger (2014) refer to them as *collgrams*. They make it possible not to rely exclusively on the frequency of the bigram in the corpus, which can be misleading because a bigram may be observed frequently, not because of its phraseological nature, but because it is made up of very frequent words (Bestgen, 2018). Conversely, a relatively rare bigram, composed of rare words, may be typical of the language. Since word frequency is already accounted for by the corpus frequency features, it was desirable to employ indices that reduce the impact of this factor. Originating in works in lexicography and foreign language learning (Church and Hanks, 1990; Durrant and Schmitt, 2009; Bestgen, 2017, 2019), they have recently shown their usefulness in predicting the lexical complexity of multi-word expressions (Bestgen, 2021). In the present case, it is assumed that these indices can serve as a proxy of the next word predictability (Kliegl et al., 2004).

Parameters	Run 1	Run 2
bagging_fraction	0.66	0.70
bagging_freq	5	5
feature_fraction	0.09	0.85
learning_rate	0.0095	0.0050
max_depth	11	no limit
max_bin	64	64
min_data_in_bin	2	5
max_leaves	11	30
min_data_in_leaf	7	5
n_iter	4800	(see text)

Table 1: LightGBM parameters for the first two runs.

Feature coverage. Some words to predict are not present in these lists and the corresponding score is thus missing. Based on the complete dataset provided by the organizers, it happens in:

- 1% (Google ngram) to 17% (Facebook and Twitter) of the tokens for the corpus frequency features,
- 9% for the ELP Lexical Characteristics, but a few features have as much as 41% missing values,
- 11% for the ELP Behavioral Measures,
- 18% for the Bigram AMs.

In total, sixteen tokens have missing values for all these features (Corpus Frequency, Lexical Characteristics and Behavioral Measures from ELP, and Bigram Association Measures). These tokens have however received values for the length and position features. All the missing values were handled by LightGBM default procedure.

4 Analyses and Results

4.1 Models Submitted to the Challenge

During the test phase, teams were allowed to submit three runs. My three submissions were all based on the features described above, the only differences between them resulting from changes in the LigthGBM parameters. They were set at their default values except those shown in Table 1. The official performances of the top five challenge submissions are given in Table 2.

Team	Run	Mean	nFix	FFD	GPT	TRT	fixProp
LAST	3	**3.8134**	3.879	**0.655**	2.197	1.524	**10.812**
LAST	2	3.8159	3.886	0.655	2.199	1.523	10.817
TALEP	1	3.8328	**3.761**	0.662	**2.180**	**1.486**	11.076
LAST	1	3.8664	3.943	0.662	2.237	1.545	10.944
TorontoCL	2	3.9287	3.944	0.671	2.227	1.516	11.286

Table 2: Performance (MAE) for the five best runs submitted to the challenge. Best scores are bolded.

The first submission was based on the parameters selected during the development phase. They were identical for the five DVs. For the other two submissions, a random grid search coded in python was used to try optimizing the parameters independently for each DV. The parameter space for this first random search is provided in Appendix A. As the measure of the challenge is the MAE averaged across the five DVs and as the system MAE for fixProp was up to 15 times higher than that of the other DVs, the optimized parameters for this variable were selected. Additional analyzes showed that they also made it possible to improve performance on the four other DVs. Their values are given in Table 1. Certain initial choices were only slightly modified. The value of other parameters such as the maximum number of leaves and the feature fraction were markedly increased, suggesting that the risk of overfit was relatively low (see https://lightgbm.readthedocs.io/en/latest/Parameters-Tuning.html). In this system, the number of iterations was optimized (thanks to the early stopping procedure) for each DV and sets at the fourth highest value: 3,740 for nFix, 3,829 for TRT, 2,861 for GPT, 3,497 for FFD, and 3,305 for fixProp.

For the third submission, a new round of random optimization was conducted by evaluating parameter values close to those selected for Run 2, independently for each DV. As it only got slightly better performance than Run 2, these parameter values are not shown to save space.

As shown in Table 2, Runs 2 and 3 ranked at the first 2 places of the challenge. This result was largely due to their better performance for fixProp since the TALEP system, second in the challenge, achieved significantly better performance for three of the five DVs, but these have less impact on the official measurement. An analysis, carried out after the end of the challenge, showed that the system would not have been more effective (average MAE of 3.8138) if, during the first optimization step, a specific model for each DV had been selected.

Using Pearson's linear correlation coefficient as a measure of effectiveness, which is unaffected by the differences in means and variability between the five DVs, Run 3 obtains an average r of 0.812 on the test set (min = 0.792 for GPT; max = 0.838 for fixProp). This value is relatively high, but it can only really be interpreted by taking into account the reliability of the average real eye-tracking feature values.

4.2 Feature Usefulness

The first part of Table 3 presents the main results of an ablation procedure aimed at examining the impact of the different types of features on the system performance. It gives the average MAE as well as the difference in percentage between each system and the best run for the average MAE and for the five DVs. It must be first stressed that all features based on lemmas and POS-tag, the two Glasgow norms and the length of the token that follows the target are useless for predicting the test set since without them the system achieves a MAE of 3.8134. They are thus discarded in all the ablation analyses. The target's positions in the sentence and the length features are clearly essential. Among the features resulting from corpora and behavioral data, it is the bigram association measures and the frequencies in the corpora that are the most useful.

Generally speaking, the feature sets have comparable utility for all DVs. However, we observe that the position in the sentences is particularly important for predicting GPT while the length of the target is more useful for nFix.

The second part of Table 3 presents an analysis of the utility of optimizing the LightGBM parameters, based on the best system. Optimizing RMSE instead of MAE is especially penalizing for GPT. Using the default values of the LightGBM param-

Models	MAE	%MAE	%nFix	%FFD	%GPT	%TRT	%fixProp
W/o behavioral data	3.849	-0.93	**-0.69**	**-1.30**	-0.75	-0.78	-1.05
W/o ELP charact.	3.859	-1.19	**-0.54**	-1.36	-0.95	-0.59	**-1.55**
W/o frequencies	3.880	-1.74	**-1.38**	-1.68	**-1.88**	-1.55	-1.87
W/o bigram AM	3.881	-1.78	-2.05	**-2.32**	**-1.39**	-1.94	-1.70
W/o length feat.	3.979	-4.35	**-5.95**	**-2.92**	-3.17	-4.43	-4.08
W/o position feat.	4.095	-7.39	-7.68	-4.44	**-22.88**	-7.48	**-4.30**
RMSE optimization	3.847	-0.87	-0.43	**0.46**	**-4.73**	-0.09	-0.43
Default Param + MAE	3.902	-2.32	-2.34	**-1.54**	**-3.52**	-2.12	-2.15
Default Param + RMSE	4.141	-8.59	-7.67	-7.65	**-12.62**	**-7.43**	-8.31
Linear Regression	4.268	-10.64	-9.04	**-7.88**	**-24.09**	-9.47	-8.26
LGBM on Length + Position	4.219	-10.63	-10.7	-11.4	**-8.18**	**-12.1**	-10.85

Table 3: Performance (MAE) of different system versions and deviation (%) from the best run ($MAE = 3.813$). Minimum and maximum values across DVs for each row are bolded.

eters is particularly penalizing when RMSE is the criterion.

A final question concerns the benefits of employing LightGBM instead of another regression algorithm when the proposed features are used. To try to provide at least a partial answer, I trained a multiple linear regression model on the basis of the features used, while adding for each feature, for which the calculation was possible, a second feature containing the logarithm of the initial value. I replaced the missing data with 0, which is probably not optimal. A stepwise regression procedure with a threshold to enter sets at $p = 0.01$ and a threshold to exit sets at $p = 0.05$ was employed to construct for each DV a model on the learning set and apply it to the test set. The results obtained are given in the second to last row of Table 3. The performances are clearly less good. It is even worse than the performance level of a LightGBM model based only on the length and position features (see the last row of Table 3). This regression system would have been ranked 10th in the challenge.

5 Conclusion

The system proposed for the 2021 CMCL Shared Task on Eye-Tracking Data Prediction was particularly effective, obtaining the first place in the challenge, but it must be kept in mind that the system that came second is superior to it for three of the five DVs. The analyzes carried out to understand its pros and cons indicate that optimizing the LightGBM parameters is quite beneficial to it as well as the different sets of features derived from corpora and behavioral data, including bigram AMs which, to my knowledge, have never been employed for this type of task.

It would have been interesting to relate these observations to the psycholinguistic literature on the factors that influence eye fixations, but this is unfortunately not possible here, for lack of space. In addition, this would first require deepening the ablation analyzes by simultaneously considering several feature sets. For instance, the lack of usefulness of the POS-tags could simply result from the links (at least partial) between them and the frequency and length of the tokens. Likewise, some of the bigram AMs are relatively sensitive to the frequency of the words that compose them (e.g., the t-score favors frequent bigrams which are usually composed of frequent words). It is thus highly probable that some of the features in the different sets (frequencies, behavioral data...) are redundant and can be removed without impairing the performance of the system. This is a potential development path.

Acknowledgements

The author wishes to thank the organizers of this shared task for putting together this valuable event and the reviewers for their very constructive comments. He is a Research Associate of the Fonds de la Recherche Scientifique - FNRS (Fédération Wallonie Bruxelles de Belgique). Computational resources were provided by the supercomputing facilities of the UCLouvain (CISM/UCL) and the Consortium des Equipements de Calcul Intensif en Fédération Wallonie Bruxelles (CECI).

References

David A. Balota, Melvin J. Yap, Keith A. Hutchison, Michael J. Cortese, Brett Kessler, Bjorn Loftis, James H. Neely, Douglas L. Nelson, Greg B. Simpson, and Rebecca Treiman. 2007. The English lexicon project. *Behavior Research Methods*, 39:445–459.

Godelieve L. M. Berry-Rogghe. 1973. The computation of collocations and their relevance in lexical studies. In Adam J Aitken, Richard W. Bailey, and Neil Hamilton-Smith, editors, *The Computer and Literary Studies*. Edinburgh University Press.

Yves Bestgen. 2017. Beyond single-word measures: L2 writing assessment, lexical richness and formulaic competence. *System*, 69:65–78.

Yves Bestgen. 2018. Evaluating the frequency threshold for selecting lexical bundles by means of an extension of the Fisher's exact test. *Corpora*, 13:205–228.

Yves Bestgen. 2019. Evaluation de textes en anglais langue ètrangère et séries phraséologiques : comparaison de deux procédures automatiques librement accessibles. *Revue française de linguistique appliquée*, 24:81–94.

Yves Bestgen. 2021. LAST at SemEval-2021 Task 1: improving multi-word complexity prediction using bigram association measures. In *Proceedings of SemEval-2021*.

Yves Bestgen and Sylviane Granger. 2014. Quantifying the development of phraseological competence in L2 English writing: An automated approach. *Journal of Second Language Writing*, 26:28–41.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Ted Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

Philip Durrant and Norbert Schmitt. 2009. To what extent do native and non-native writers make use of collocations? *International Review of Applied Linguistics in Language Teaching*, 47:157–177.

Stefan Evert. 2009. Corpora and collocations. In Anke Lüdeling and Merja Kytö, editors, *Corpus Linguistics. An International Handbook*, pages 1211–1248. Mouton de Gruyter.

Amac Herdagdelen and Marco Marelli. 2017. Social media and language processing: How Facebook and Twitter provide the best frequency estimates for studying word recognition. *Cognitive Science*, 41:976–995.

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 the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 538–549, Hong Kong, China. Association for Computational Linguistics.

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. 5:180291*, 5(180291):1–13.

Nora Hollenstein, Marius Troendle, Ce Zhang, and Nicolas Langer. 2020. ZuCo 2.0: A dataset of physiological recordings during natural reading and annotation. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 138–146. European Language Resources Association.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3146–3154. Curran Associates, Inc.

Adam Kilgarriff, Vít Baisa, Jan Bušta, Miloš Jakubíček, Vojtěch Kovář, Jan Michelfeit, Pavel Rychlý, and Vít Suchomel. 2014. The Sketch Engine: ten years on. *Lexicography*, 1(1):7–36.

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*, 16:262–284.

Kristopher Kyle, Scott Crossley, and Cynthia Berger. 2018. The tool for the automatic analysis of lexical sophistication (TAALES): version 2.0. *Behavior Research Methods*, 50:1030–1046.

Abhijit Mishra and Pushpak Bhattacharyya. 2018. *Applications of Eye Tracking in Language Processing and Other Areas*, pages 23–46. Springer Singapore, Singapore.

Pavel Pecina. 2010. Lexical association measures and collocation extraction. *Language Resources & Evaluation*, 44:137–158.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, pages 44–49.

Graham G. Scott, Anne Keitel, Marc Becirspahic, Bo Yao, and Sara C. Sereno. 2019. The Glasgow norms: Ratings of 5,500 words on nine scales. *Behavior Research Methods*, 51:1258–1270.

Cyrus Shaoul and Westbury Chris. 2006. USENET orthographic frequencies for 111,627 English words.

Matthew Shardlow, Richard Evans, Gustavo Paetzold, and Marcos Zampieri. 2021. SemEval-2021 task 1: Lexical complexity prediction. In *Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2021)*.

Tal Yarkoni, David A. Balota, and Melvin Yap. 2008. Moving beyond Coltheart's N: A new measure of orthographic similarity. *Psychonomic Bulletin & Review*, 16:971–979.

A Appendix

At the request of a reviewer, the parameter space for the first random search is provided below. Those for the second random search are not provided as they did not allow to really improve the performances.

```
param_grid = {
 'max_bin': [16,32,48,64,80,96,112,128,
    160,192,224,256],
 'min_data_in_bin':[2,3,4,5,6,8,10,12,
    15,20],
 'num_leaves': [4,5,6,7,8,9,10,11,12,13,
    15,18,21,25,30],
 'learning_rate': [0.005,0.007,0.009,
    0.011,0.014,0.018,0.022,0.026,0.03,
    0.035,0.05],
 'min_data_in_leaf': [2,3,4,5,6,7,8,9,10,
    11,12,13,15,18,21,25,30],
 'max_depth': [3,4,5,6,7,8,9,10,11,12,13,
    -1],
 'feature_fraction': list(np.linspace(
    0.01, 0.90, 91)),
 'bagging_freq': list(range(3, 7, 1)),
 'bagging_fraction': list(np.linspace(
    0.50, 0.90, 9))
}
```