TorontoCL at CMCL 2021 Shared Task: RoBERTa with Multi-Stage Fine-Tuning for Eye-Tracking Prediction

Bai Li[1,2], **Frank Rudzicz**[1,2,3]
[1] University of Toronto, Department of Computer Science
[2] Vector Institute for Artificial Intelligence
[3] Unity Health Toronto
`{bai, frank}@cs.toronto.edu`

Abstract

Eye movement data during reading is a useful source of information for understanding language comprehension processes. In this paper, we describe our submission to the CMCL 2021 shared task on predicting human reading patterns. Our model uses RoBERTa with a regression layer to predict 5 eye-tracking features. We train the model in two stages: we first fine-tune on the Provo corpus (another eye-tracking dataset), then fine-tune on the task data. We compare different Transformer models and apply ensembling methods to improve the performance. Our final submission achieves a MAE score of 3.929, ranking 3rd place out of 13 teams that participated in this shared task.

1 Introduction

Eye-tracking data provides precise records of where humans look during reading, with millisecond-level accuracy. This type of data has recently been leveraged for uses in natural language processing: it can improve performance on a variety of downstream tasks, such as part-of-speech tagging (Barrett et al., 2016), dependency parsing (Strzyz et al., 2019), and for cognitively-inspired evaluation methods for word embeddings (Søgaard, 2016). Meanwhile, Transformer-based language models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) have been applied to achieve state-of-the-art performance on many natural language tasks. The CMCL 2021 shared task aims to add to our understanding of how language models can relate to eye movement features.

In this paper, we present our submission to this shared task, which achieves third place on the leaderboard. We first explore some simple baselines using token-level features, and find that these are already somewhat competitive with the final model's performance. Next, we describe our model architecture, which is based on RoBERTa (Figure

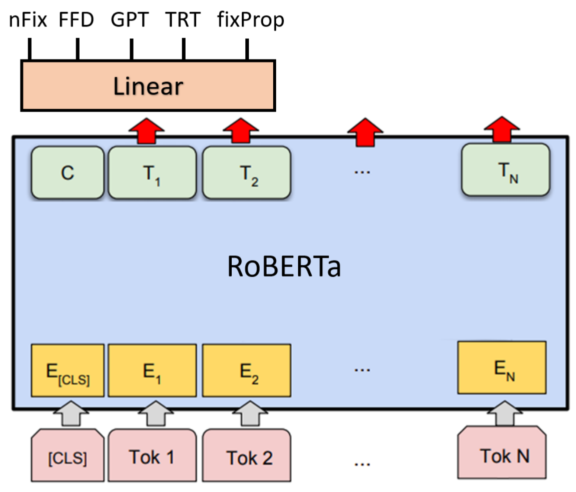

Figure 1: Our model consists of RoBERTa with a regression head on each token, which is a linear layer that predicts the 5 output features from the last layer's embeddings. The model is initialized from pretrained weights and fine-tuned on the task data.

1). We find that model ensembling offers a substantial performance gain over a single model. Finally, we augment the provided training data with the publicly available Provo eye-tracking corpus and combine them using a two-stage fine-tuning procedure; this results in a moderate performance gain. Our source code is available at `https://github.com/SPOClab-ca/cmcl-shared-task`.

2 Task Description

The shared task format is described in Hollenstein et al. (2021), which we will briefly summarize here. The task data consists of sentences derived from the ZuCo 1.0 and ZuCo 2.0 datasets; 800 sentences (15.7 tokens) were provided as training data and 191 sentences (3.5k tokens) were held out for evaluation. The objective is to predict five eye-tracking features for each token:

- Number of fixations on the current word (*nFix*).

Model	nFix	FFD	GPT	TRT	fixProp	All (Dev)
Median	7.208	1.162	3.547	2.732	21.179	7.165
Linear regression	4.590	0.795	2.995	1.812	13.552	4.749
SVR (RBF kernel)	4.440	0.723	2.728	1.728	12.077	4.339

Table 1: Baseline results: 'Median' is a model that always predicts the median of the training data; the linear regression and SVR models use 4 token-level surface features described in Section 3.1.

Model	nFix	FFD	GPT	TRT	fixProp	All (Dev)
BERT-base	4.289	0.704	2.645	1.678	11.155	4.094
BERT-large	4.150	0.682	2.493	1.616	11.013	3.991
RoBERTa-base	**4.066**	**0.681**	**2.443**	**1.570**	**10.981**	**3.930**
RoBERTa-large	4.156	0.681	2.468	1.623	11.047	3.995

Table 2: MAE using BERT and RoBERTa models with fine-tuning.

- First fixation duration of the word (*FFD*).
- Go-past time: the time from the first fixation of a word until the first fixation beyond it (*GPT*).
- Total reading time of all fixations of the word, including regressions (*TRT*).
- Proportion of participants that fixated on the word (*fixProp*).

The features are averages across multiple participants, and each feature is scaled to be in the range [0, 100]. The evaluation metric is the mean absolute error (MAE) between the predicted and ground truth values, with all features weighted equally.

Since each team is allowed only a small number of submissions, we define our own train and test split to compare our models' performance during development. We use the first 600 sentences as training data and the last 200 sentences for evaluation during development. Except for the submission results (Table 4), all experimental results reported in this paper are on this development train/test split.

3 Our Approach

3.1 Baselines

We start by implementing some simple baselines using token-level surface features. Previous research in eye tracking found that longer words and low-frequency words have higher probabilities of being fixated upon (Rayner, 1998). We extract the following features for each token:

- Length of token in characters.
- Log of the frequency of token in English text, retrieved using the `wordfreq`[1] library.
- Boolean of whether token contains any uppercase characters.
- Boolean of whether token contains any punctuation.

Using these features, we train linear regression and support vector regression models separately for each of the 5 output features (Table 1). Despite the simplicity of these features, which do not use any contextual information, they already perform much better than the median baseline. This indicates that much of the variance in all 5 eye-tracking features are explained by surface-level cues.

3.2 Fine-tuning Transformers

Our main model uses RoBERTa (Liu et al., 2019) with a linear feedforward layer to predict the 5 output features simultaneously from the last hidden layers of each token. In cases where the original token is split into multiple RoBERTa tokens, we use the first RoBERTa token to make the prediction. The model is initialized with pretrained weights and fine-tuned on the task data to minimize the sum of mean squared errors across all 5 features.

As the task data is relatively small, we found that the model needs to be fine-tuned for 100-150 epochs to reach optimal performance, far greater than the recommended 2-4 epochs (Devlin et al., 2019). We trained the model using the AdamW optimizer (Loshchilov and Hutter, 2018) with learning rates of {1e-5, 2e-5, 5e-5, 1e-4} and batch sizes of {8, 16, 32}; all other hyperparameters were left

[1] `https://github.com/LuminosoInsight/wordfreq/`

Model	nFix	FFD	GPT	TRT	fixProp	All (Dev)
Single Model	4.066	0.681	2.443	1.570	10.891	3.930
Ensemble of 2	3.978	0.671	2.350	1.534	10.714	3.849
Ensemble of 5	3.944	0.669	2.321	**1.521**	10.665	3.824
Ensemble of 10	**3.943**	**0.666**	**2.316**	1.522	**10.660**	**3.821**

Table 3: Ensembles of RoBERTa-base model, obtained by taking a simple mean of the predictions of individual models. This improves our overall performance by about 0.09 MAE compared to a single model, but with diminishing returns past 5 models.

Training Data	nFix	FFD	GPT	TRT	fixProp	All (Dev)	Submission
Task Only (Single)	4.066	0.681	2.443	1.570	10.891	3.930	n/a
Provo + Task (Single)	3.984	0.713	2.424	1.556	10.781	3.892	n/a
Task Only (Ensemble)	3.943	0.666	2.316	1.522	10.660	3.821	3.974
Provo + Task (ensemble)	**3.888**	**0.664**	**2.306**	**1.499**	**10.586**	**3.789**	**3.929**

Table 4: Comparison of model trained using the provided versus two-stage fine-tuning using Provo data. The additional pretraining improved overall performance by about 0.04 MAE. Our best submission is an ensemble of 10 RoBERTa-base models with two-stage fine-tuning.

at their default settings using the HuggingFace library (Wolf et al., 2020).

In addition to RoBERTa, we experiment with BERT (Devlin et al., 2019); we try both the base and large versions of BERT and RoBERTa, using a similar range of hyperparameters for each (Table 2). RoBERTa-base performed the best in our validation experiments; surprisingly, RoBERTa-large had worse performance.

3.3 Model Ensembling

We use a simple approach to ensembling: we train multiple versions of an identical model using different random seeds and make predictions on the test data. These predictions are the averaged to obtain the final submission. In our experiments (Table 3), ensembling greatly improves our performance, but with diminishing returns: the MAE of the 10-model ensemble is only marginally better than the 5-model ensemble. We use ensembles of 10 models in our final submission.

3.4 Domain Adaptation from Provo

In addition to the task data provided, we also use data from the Provo corpus (Luke and Christianson, 2018). This corpus contains eye-tracking data from 84 participants reading 2.6k words from a variety of text sources. The corpus also provides predictability norms and extracted syntactic and semantic features for each word, which we do not use.

We process the Provo data to be similar to the task data so that they can be combined. First, we identify the Provo features that are most similar to each of the output features: we map *IA_FIXATION_COUNT* to *nFix*, *IA_FIRST_FIXATION_DURATION* to *FFD*, *IA_REGRESSION_PATH_DURATION* to *GPT*, and *IA_DWELL_TIME* to *TRT*, taking the mean across all participants for each feature. For the *fixProp* feature, we calculate the proportion of participants where $IA\_DWELL\_TIME > 0$ for each word. Finally, we scale all five features to have the same mean and standard deviation as the task data, and verify that their distributions and pairwise scatterplots are similar (Figure 2).

We use two-stage fine-tuning to combine the Provo data with the task data. In two-stage fine-tuning, the entire model is fine-tuned on an auxiliary task before fine-tuning on the target task – this often yields a performance improvement, especially when the target task has a small amount of data (Pruksachatkun et al., 2020). In our case, we fine-tune the RoBERTa-base model for 100 epochs on the Provo data, then fine-tune for another 150 epochs on the task data. This gave a considerable improvement on both the development and submission scores (Table 4). Our best final submission is an ensemble of 10 identical models trained this way with different random seeds.

4 Conclusion and Future Work

We propose a simple approach to predict eye-tracking features using the RoBERTa model cus-

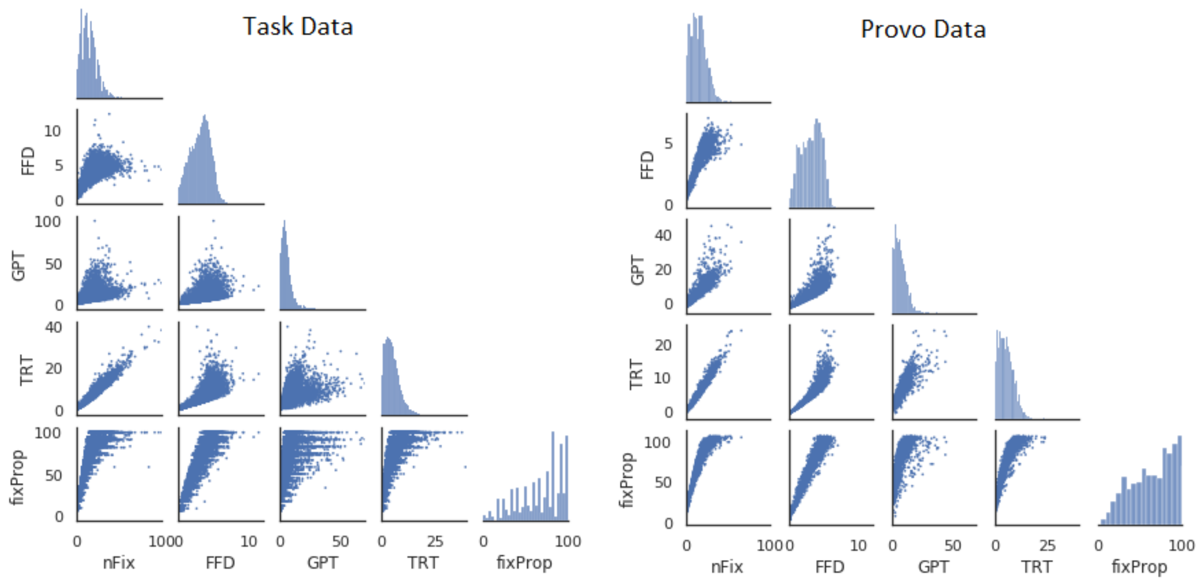

Figure 2: Distributions and pairwise scatterplots of the task data (left) and Provo data processed to match the mean and standard deviation of the task data (right).

tomized with a per-token regression head. Our initial model uses the standard fine-tuning procedure; experiments show that the performance is further improved by model ensembling and domain adaptation by two-stage fine-tuning on an intermediate eye-tracking task. Our best model achieves third place on the leaderboard.

In future work, several avenues may be explored to further improve performance. First, we did not combine our feature engineering baseline with the RoBERTa model – engineered features (such as frequency statistics or neurolinguistic norms) would provide the model with information not contained in RoBERTa. Second, we only experimented with a small subset of features from the Provo corpus for domain adaptation, whereas it is not actually necessary for the auxiliary fine-tuning task to match the target task. Thus, it may be possible to achieve better performance by fine-tuning on a different set of Provo features, or a different dataset entirely.

Acknowledgements

We thank Dr Jeanne Sinclair for her helpful discussions during this project. FR is supported by a CIFAR Chair in Artificial Intelligence.

References

Maria Barrett, Joachim Bingel, Frank Keller, and Anders Søgaard. 2016. Weakly supervised part-of-speech tagging using eye-tracking data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 579–584.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

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*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

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.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel Bowman. 2020. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of the 58th Annual*

Meeting of the Association for Computational Linguistics, pages 5231–5247.

Keith Rayner. 1998. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124(3):372.

Anders Søgaard. 2016. Evaluating word embeddings with fMRI and eye-tracking. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 116–121.

Michalina Strzyz, David Vilares, and Carlos Gómez-Rodríguez. 2019. Towards making a dependency parser see. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1500–1506.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.