Fine-Tuning Pre-Trained Language Models with Gaze Supervision

Shuwen Deng[1], **Paul Prasse**[1], **David R. Reich**[1], **Tobias Scheffer**[1], **Lena A. Jäger**[1,2]
[1] Department of Computer Science, University of Potsdam, Germany
[2] Department of Computational Linguistics, University of Zurich, Switzerland
{deng, prasse, david.reich, tobias.scheffer}@uni-potsdam.de
jaeger@cl.uzh.ch

Abstract

Human gaze data provide cognitive information that reflect human language comprehension, and has been effectively integrated into a variety of natural language processing (NLP) tasks, demonstrating improved performance over corresponding plain text-based models. In this work, we propose to integrate a gaze module into pre-trained language models (LMs) at the fine-tuning stage to improve their capabilities to learn representations that are grounded in human language processing. This is done by extending the conventional purely text-based fine-tuning objective with an auxiliary loss to exploit cognitive signals. The gaze module is only included during training, retaining compatibility with existing pre-trained LM-based pipelines. We evaluate the proposed approach using two distinct pre-trained LMs on the GLUE benchmark and observe that the proposed model improves performance compared to both standard fine-tuning and traditional text augmentation baselines. Our code is publicly available.[1]

1 Introduction

As humans read, the unconscious cognitive processes that unfold in their minds while comprehending the stimulus text are reflected in their eye movement behavior (Just and Carpenter, 1980). These gaze signals hold the potential to enhance NLP tasks. Research has focused on using aggregated word-level gaze features to enrich text features (Barrett et al., 2016; Mishra et al., 2016; Hollenstein and Zhang, 2019) or to regularize neural attention mechanisms, making their inductive bias more human-like (Barrett et al., 2018; Sood et al., 2020, 2023).

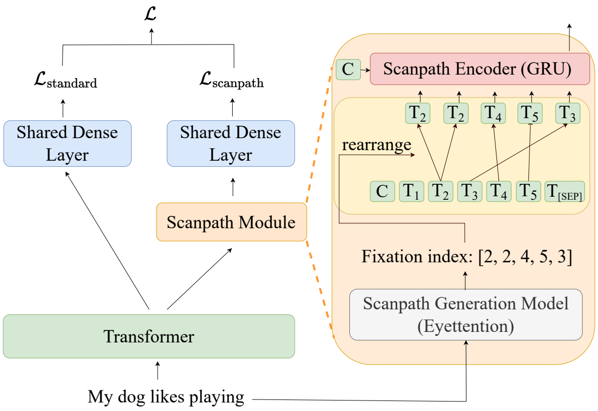
$\mathcal{L}$
$\mathcal{L}_{\text{standard}}$
$\mathcal{L}_{\text{scanpath}}$
Shared Dense Layer
Shared Dense Layer
Scanpath Module
Transformer
My dog likes playing
C
Scanpath Encoder (GRU)
T_2 T_2 T_4 T_5 T_3
rearrange
C T_1 T_2 T_3 T_4 T_5 $T_{[SEP]}$
Fixation index: [2, 2, 4, 5, 3]
Scanpath Generation Model (Eyettention)

Figure 1: Overall architecture during training. The standard objective is augmented with an auxiliary loss from a scanpath-integrated branch, where token embeddings are rearranged based on the simulated fixation sequence.

Moreover, there has been growing interest in adopting non-aggregated scanpaths (i.e., sequences of consecutive fixations) to augment LMs. These scanpaths capture the complete sequential ordering of a reader's gaze behavior and approximate their attention. Mishra et al. (2017) and Khurana et al. (2023) employed neural networks to independently encode scanpaths and text, followed by the fusion of the features extracted from both modalities. Yang and Hollenstein (2023) proposed rearranging the contextualized token embeddings produced by pre-trained LMs based on the order in which the reader fixates on the words, followed by applying sequence modeling to the reordered sequence. To tackle the issue of gaze data scarcity, Deng et al. (2023a) explored the possibility of augmenting LMs using synthetic scanpaths, generated by a scanpath generation model. Remarkably, synthetic scanpaths demonstrated advantages across various NLP tasks, particularly in settings with limited labeled examples for the downstream task.

In contrast to previous studies that concentrated on learning joint cross-modal representations of text and scanpath, we start from a different perspective and explore utilizing gaze data to improve on

[1] https://github.com/aeye-lab/ACL-GazeSupervisedLM

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 217–224
August 11-16, 2024 ©2024 Association for Computational Linguistics

the learned text representations of pre-trained LMs during the fine-tuning stage, without incurring additional computational effort when using the model at application time. To this end, we extend the standard pre-trained LM fine-tuning objective with an auxiliary loss by integrating a scanpath module, which serves a dual purpose. First, the auxiliary loss can effectively incorporate human-like gaze signals generated using a scanpath generation model and thus provide informative gradients to guide the LM towards more representative local minima. Second, reordering the token-embedding sequence based on the fixation sequence can diversify textual information, potentially improving generalization performance (Xie et al., 2020). This stands in contrast to heuristic text augmentation strategies, like random word insertion, replacement, swapping, and deletion (Wei and Zou, 2019; Xie et al., 2020). Scanpaths inherently contain cognitive information that better aligns with and complements textual information.

Notably, our proposed gaze module is only active during training (fine-tuning), ensuring alignment with the standard usage of LMs after this stage. This offers two key benefits. First, it facilitates seamless integration with existing LM-based pipelines. Second, at deployment time, it eliminates the need to either collect real-time gaze recordings, which is costly and impractical for most use-cases, or generate synthetic gaze data, which is often computationally challenging for devices with limited computational resources.

On the General Language Understanding Evaluation (GLUE) benchmark, our proposed gaze-augmented fine-tuning outperforms both standard text-only fine-tuning and traditional text augmentation baselines, without incurring additional computational effort at application time.

2 Method

In this section, we start out with a brief description of the conventional fine-tuning procedure for Transformer-based encoders on downstream tasks. Subsequently, we introduce our method, and explain how it incorporates synthetic scanpaths into this fine-tuning procedure to enhance representation learning of Transformer-based encoders. The overall model architecture is illustrated in Figure 1.

Preliminaries Our learning objective is to solve standard multi-class classification or regression problems. We assume access to a Transformer-based pre-trained LM like BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019). In the conventional fine-tuning approach for downstream tasks, the pre-trained LM is adapted to a specific task by fine-tuning all the parameters end-to-end using task-specific inputs and outputs. The final hidden state of the "[CLS]" token typically serves as the aggregated sentence representation, which is then fed into a newly initialized (series of) dense layer(s) with output neurons corresponding to the number of labels in the task. We minimize the standard cross-entropy loss for classification and mean-squared-error loss for regression, denoted as $\mathcal{L}_{\text{standard}}$ in Figure 1.

Scanpath Integration We extend the standard fine-tuning framework by integrating a scanpath module. The design of the scanpath module follows the prior work of Deng et al. (2023a) and Yang and Hollenstein (2023). Specifically, the Transformer encoder produces contextualized token embeddings for a given sentence, with each embedding associated with its position index in the sequence. Simultaneously, a synthetic scanpath (fixation-index sequence) is generated based on the same sentence using the scanpath-generation model Eyettention (Deng et al., 2023b), which has demonstrated effectiveness in simulating human-like scanpaths during reading (see Appendix A for detailed information about the Eyettention model). The scanpath module then rearranges the token-embedding sequence based on the simulated fixation sequence. Subsequently, we use a scanpath encoder, implemented as a layer of Gated Recurrent Units (GRU), to process the reordered sequence. The output from the last step of the scanpath encoder is then forwarded to the subsequent dense layer. For the branch that takes the scanpath into account, we introduce an additional loss term, referred to as $\mathcal{L}_{\text{scanpath}}$ in Figure 1, which represents the cross-entropy loss for classification and the mean-squared-error loss for regression.

Training Objective We combine the standard purely text-based loss and the scanpath-integrated loss with a trade-off factor λ. The final training objective is defined as:

$$\mathcal{L} := \mathcal{L}_{\text{standard}} + \lambda \mathcal{L}_{\text{scanpath}}.$$

The joint optimization of the two branches facilitates the flow of cognitive information from the scanpath module to the Transformer through back-propagation, thereby improving its capability to

K	Model	MNLI	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	Avg
200	BERT	$42.10_{0.46}$	$62.16_{1.30}$	$73.58_{0.56}$	$77.68_{1.71}$	$18.52_{4.24}$	$80.48_{0.32}$	$82.12_{0.43}$	$54.95_{0.67}$	61.45
	+EDA	$\mathbf{47.74}_{1.10}$	$\mathbf{64.89}_{0.56}$	$\mathbf{76.23}_{0.34}$	$80.48_{1.26}$	$14.05_{2.84}$†	$79.56_{0.62}$†	$82.68_{0.40}$	$55.74_{0.30}$	62.67
	+SP	$42.63_{0.82}$	$64.47_{0.84}$	$73.83_{0.44}$	$\mathbf{81.19}_{0.98}$	$\mathbf{23.33}_{3.42}$	$\mathbf{82.01}_{0.28}$	$\mathbf{82.71}_{0.48}$	$\mathbf{56.10}_{0.67}$	**63.28**
500	BERT	$52.35_{1.23}$	$67.33_{0.29}$	$77.78_{0.46}$	$84.17_{0.28}$	$30.29_{1.86}$	$83.90_{0.24}$	$83.15_{0.26}$	$60.43_{1.07}$	67.43
	+EDA	$\mathbf{56.37}_{0.88}$	$\mathbf{68.03}_{0.33}$	$\mathbf{78.48}_{0.32}$	$\mathbf{85.37}_{0.17}$	$28.89_{1.58}$†	$83.28_{0.24}$†	$84.00_{0.28}$	$60.43_{0.49}$	68.11
	+SP	$55.40_{0.61}$	$67.86_{0.42}$	$78.19_{0.24}$	$84.22_{0.52}$	$\mathbf{35.87}_{1.50}$	$\mathbf{85.26}_{0.29}$	$\mathbf{84.52}_{0.46}$	$\mathbf{61.44}_{0.43}$	**69.10**
1000	BERT	$60.51_{0.66}$	$69.40_{0.54}$	$79.53_{0.16}$	$85.25_{0.51}$	$39.92_{0.86}$	$86.22_{0.11}$	$85.42_{0.23}$	$63.10_{1.16}$	71.17
	+EDA	$61.58_{0.50}$	$69.91_{0.35}$	$\mathbf{80.49}_{0.16}$	$86.10_{0.34}$	$31.04_{1.89}$†	$85.50_{0.22}$†	$86.37_{0.44}$	$\mathbf{64.26}_{1.16}$	70.66†
	+SP	$\mathbf{61.75}_{0.32}$	$\mathbf{70.58}_{0.30}$	$80.24_{0.33}$	$\mathbf{86.70}_{0.09}$	$\mathbf{42.45}_{0.59}$	$\mathbf{86.73}_{0.14}$	$\mathbf{86.77}_{0.69}$	$63.18_{1.08}$	**72.3**
200	RoBERTa	$40.06_{0.68}$	$68.59_{0.54}$	$77.21_{0.60}$	$\mathbf{88.56}_{0.39}$	$\mathbf{30.29}_{2.55}$	$82.84_{0.43}$	$83.37_{0.16}$	$55.81_{1.15}$	65.84
	+EDA	$\mathbf{53.64}_{0.44}$	$68.84_{0.71}$	$77.52_{0.57}$	$87.94_{0.64}$†	$23.30_{4.16}$†	$\mathbf{83.86}_{0.10}$	$\mathbf{84.05}_{0.49}$	$58.41_{1.20}$	**67.20**
	+SP	$44.90_{0.63}$	$\mathbf{69.05}_{0.69}$	$\mathbf{78.14}_{0.68}$	$87.11_{0.86}$†	$29.07_{3.18}$†	$82.42_{0.24}$†	$83.86_{0.62}$	$\mathbf{63.03}_{2.58}$	**67.20**
500	RoBERTa	$\mathbf{65.20}_{0.46}$	$73.42_{0.48}$	$81.54_{0.22}$	$89.61_{0.35}$	$\mathbf{39.59}_{0.95}$	$\mathbf{86.68}_{0.30}$	$86.09_{0.36}$	$62.24_{1.92}$	73.05
	+EDA	$64.97_{0.56}$†	$71.57_{0.45}$†	$81.20_{0.23}$†	$89.27_{0.35}$†	$36.05_{2.28}$†	$86.46_{0.26}$†	$\mathbf{87.49}_{0.67}$	$59.49_{1.55}$†	72.06†
	+SP	$64.89_{0.42}$†	$\mathbf{73.79}_{0.30}$	$\mathbf{81.78}_{0.16}$	$\mathbf{89.75}_{0.30}$	$39.07_{1.96}$†	$86.29_{0.07}$†	$87.00_{0.54}$	$\mathbf{68.01}_{1.07}$	**73.82**
1000	RoBERTa	$\mathbf{70.91}_{0.61}$	$\mathbf{75.63}_{0.29}$	$83.43_{0.12}$	$\mathbf{90.69}_{0.24}$	$\mathbf{44.78}_{0.65}$	$88.06_{0.19}$	$88.85_{0.19}$	$64.91_{1.26}$	75.91
	+EDA	$70.84_{0.34}$†	$74.59_{0.52}$†	$82.64_{0.47}$†	$90.23_{0.38}$†	$41.44_{1.18}$†	$87.79_{0.15}$†	$\mathbf{89.60}_{0.41}$	$63.25_{2.00}$†	75.05†
	+SP	$70.69_{0.37}$†	$75.40_{0.16}$†	$\mathbf{83.59}_{0.42}$	$89.91_{0.35}$†	$44.43_{1.88}$†	$\mathbf{88.12}_{0.17}$	$89.42_{0.53}$	$\mathbf{72.71}_{0.73}$	**76.78**

Table 1: Results on the GLUE benchmark with $K = \{200, 500, 1000\}$ training instances. We use F1 for QQP and MRPC, Spearman correlation for STS-B, Matthews correlation for CoLA, and accuracy for the remaining tasks. We perform 5 runs with different random seeds and report the means along with standard errors. The dagger "†" indicates performance that is inferior to standard fine-tuning.

process and comprehend text. Consequently, during testing, we can remove the scanpath module and generate predictions solely from the Transformer and the final dense layer. This ensures alignment with standard LM usage after the fine-tuning stage, notably preserving its intrinsic efficiency and compatibility.

3 Experiments

3.1 Evaluation Setup

Data Sets We conduct experiments on the GLUE benchmark (Wang et al., 2018), including sentiment analysis (SST-2), linguistic acceptability (CoLA), similarity and paraphrase tasks (MRPC, STS-B, QQP), and natural language inference tasks (MNLI, QNLI, RTE).

Model and Data Setup We use BERT$_{\text{base}}$ (Devlin et al., 2019) and RoBERTa$_{\text{base}}$ (Liu et al., 2019) as the base models in the experiments. We primarily focus on a low-resource setting where only limited labeled examples for the downstream task are available. In such cases, effective fine-tuning strategies are crucial to enable high-capacity LMs to learn more informative representations for enhanced performance in downstream tasks (Zhang et al., 2021). For each task, we sample a small subset of training instances with sizes $K = \{200, 500, 1000\}$. We take an additional 1,000 instances from the original training set as the development set and use the original development set for testing. Additionally, we consider a high-resource setting where we use the entire training set and report the results on the GLUE development sets. Appendix B gives further details about training and hyper-parameter tuning.

Baselines We compare our proposed method with the standard text-only fine-tuning using only $\mathcal{L}_{\text{standard}}$ as the training objective. Moreover, we compare to the Easy Data Augmentation (EDA) method (Wei and Zou, 2019), which randomly performs word insertion, replacement, swap, and deletion in the text to augment the training data.

3.2 Results

Low-Resource Performance Table 1 shows that, overall, our scanpath-augmented fine-tuning (*+SP*) consistently outperforms the standard fine-tuning and EDA baselines, regardless of the number of training instances. We observe performance gains of 2-3% for BERT and 1-2% for RoBERTa over standard fine-tuning. At the per-task level, our method outperforms standard fine-tuning across all tasks in all setups for BERT, and on five, five and four out of eight tasks when trained with 200, 500, and 1,000 instances, respectively, for RoBERTa. The improvements are larger with fewer training instances, indicating the efficacy of our method in low-resource scenarios. Notably, for tasks like CoLA and STS-B, where the EDA method yields largely inferior results compared to standard fine-tuning (Model=BERT), our method shows superior performance. This suggests that the scanpath,

Model	MNLI	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	Avg
BERT	83.87	88.02	91.01	92.43	59.90	89.47	90.51	66.79	82.75
+EDA	83.82†	87.53†	90.79†	92.55	56.88†	88.67†	90.94	71.12	82.79
+SP	**84.17**	**88.27**	**91.38**	**93.23**	**64.27**	**89.61**	**91.60**	**71.48**	**84.25**
RoBERTa	87.77	89.03	92.88	94.84	61.48	**90.58**	**93.15**	77.98	85.96
+EDA	87.71†	88.58†	92.48†	95.41	58.88†	90.35†	92.93†	76.17†	85.31†
+SP	**87.95**	**89.10**	**92.97**	**94.95**	**63.20**	90.55†	92.93†	**80.14**	**86.47**

Table 2: Results on the GLUE development sets using all training samples. The dagger "†" indicates performance that is inferior to standard fine-tuning.

which inherently contains cognitive information, aligns with and complements textual information effectively.

High-Resource Performance In Table 2, we present the results of different methods when using all training instances. Our scanpath-augmented fine-tuning (+*SP*) achieves the highest overall performance. While the gains are not as significant as in the low-resource setting for most tasks, notable improvements persist for tasks like CoLA and RTE. In contrast, the EDA method fails to enhance performance over standard fine-tuning overall, which is in line with findings from previous research (Longpre et al., 2020).

3.3 Ablation Studies

Location of the Scanpath Module We explore the impact of integrating the scanpath module at different feature-representation levels on the model's performance. Specifically, we experiment with placing the scanpath module after the 11th, 8th, 5th, and embedding layer of the Transformer. In these cases, it is straightforward to use the subsequent Transformer layers to process the scanpath-guided reordered sequence; we therefore remove the scanpath encoder from the module. Moreover, we add extra positional embeddings to the token embeddings after the rearrangement, providing information about the positions of tokens in the sequence.

Table 3 shows that integrating the scanpath module into the model, regardless of its placement, yields improved performance compared to standard text-only fine-tuning. However, placing it at a lower position within the Transformer results in smaller gains. This may be attributed to the top Transformer layers capturing richer semantic information (Jawahar et al., 2019). Placing the scanpath module at the top facilitates better access to this information, potentially aiding in leveraging cognitive information. Furthermore, adding extra positional information to the reordered sequence marginally impacts performance.

Model	SST-2	CoLA	MRPC	RTE	Avg.
BERT	92.43	59.90	90.51	66.79	77.41
+SP (-AfterLayer-12)	**93.23**	**64.27**	**91.60**	71.48	**80.15**
+SP-AfterLayer-11	92.89	63.38	91.19	**71.84**	79.83
+Pos Emb	93.00	62.91	91.09	70.40	79.35
+SP-AfterLayer-8	93.12	62.44	91.36	70.04	79.24
+Pos Emb	93.12	63.04	91.00	69.68	79.21
+SP-AfterLayer-5	93.12	61.34	90.88	70.40	78.94
+Pos Emb	92.89	61.62	91.03	71.48	79.26
+SP-Emb	**93.23**	61.11	90.82	68.23	78.35

Table 3: Comparison of the *Scanpath Module* at various model locations: after the n-th Transformer layer (*SP-AfterLayer-n*), and after the Transformer's embedding layer (*SP-Emb*). We add extra positional embeddings to the token embeddings in the reordered sequence (+*Pos Emb*).

Scanpath vs Random Order The core principle of the scanpath module is to utilize the order of fixations to integrate estimated cognitive information into the model. To study whether the observed gains truly arise from the order of fixations, we compare our method which rearranges the token-embedding sequence based on the scanpath to two baselines: (1) shuffling the scanpath ordering, and (2) randomly shuffling the token-embedding sequence. Table 4 shows that shuffling the scanpath results in consistent performance drops across all tasks, indicating the importance of the order of fixations. Furthermore, excluding the scanpath and randomly shuffling BERT token embeddings leads to a large decrease in performance gain, underscoring the importance of both fixated words and their order in enhancing model performance.

Model	SST-2	CoLA	MRPC	RTE	Avg.
BERT	92.43	59.90	90.51	66.79	77.41
+SP	**93.23**	**64.27**	**91.60**	**71.48**	**80.15**
+Shuffle SP	93.00	63.81	91.34	71.12	79.82
+Random Shuffle	92.78	60.66	91.42	68.95	78.45

Table 4: Comparison of strategies for reordering token embeddings: scanpath-guided (*SP*), shuffled scanpath-guided (*Shuffle SP*), and (*Random Shuffle*).

4 Conclusion

Our work contributes to the broad effort of enriching NLP models by grounding them in various domains of experience. Specifically, we focus on the use of scanpath data, demonstrating its vital role in enhancing textual representation learning. By extending the standard pre-trained LM fine-tuning objective with a scanpath-integrated loss, we ground the LM in human language processing. Finally, our experiments show that the proposed method surpasses standard fine-tuning and EDA baselines on the GLUE benchmark, pointing to the potentially promising future direction of enriching textual representations with gaze data, especially for low-resource tasks and languages (Reich et al., 2024). However, it should be noted that the performance gains achieved by incorporating gaze supervision vary across different NLP tasks. Future work may include further analysis of the impact of incorporating cognitive information into language models on specific downstream tasks.

Limitations

One limitation of our work is that the scanpath-generation model—Eyettention—was pre-trained on a single eye-tracking corpus with a relatively small sample (see Appendix A). Participants read sentences covering only a single domain and a narrow range of text difficulty levels. This limitation may restrict the knowledge acquired by Eyettention concerning human language processing, thus potentially leading to limited benefits when integrating simulated gaze data into LMs. In our experiments, we observe that our proposed fine-tuning scheme provides smaller benefits to RoBERTa than BERT, even in the low-resource setting. The key difference between these models is the scale of unsupervised pre-training. We hypothesize that RoBERTa which is pre-trained on a larger scale of data has learnt sufficiently robust language representations, and to further improve its representation learning capability, a more competitive scanpath-generation model, trained on a large eye-tracking dataset that covers diverse domains of texts, might be required.

Furthermore, it is worth exploring the performance of the proposed approach when using other state-of-the-art scanpath generators. Different architectures have been developed recently in the field (Bolliger et al., 2023; Khurana et al., 2023). Exploring the strengths and weaknesses of different scanpath generators when integrated into LMs could provide valuable insight into the development of improved scanpath generators for benefiting NLP tasks.

Ethics Statement

It is essential to acknowledge potential privacy risks in the collection, sharing, and processing of human gaze data. Due to the highly individual nature of eye movements, there exists a possibility of extracting sensitive information such as a participant's identity (Jäger et al., 2020; Makowski et al., 2021), gender (Sammaknejad et al., 2017) and ethnicity (Blignaut and Wium, 2014) from gaze data, posing a risk of privacy leakage. The use of synthetic gaze data can help alleviate the necessity for large-scale experiments involving human subjects, although some amount of human gaze data remains necessary to train generative models.

Acknowledgements

This work was partially funded by the German Federal Ministry of Education and Research under grant 01| S20043.

References

Maria Barrett, Joachim Bingel, Nora Hollenstein, Marek Rei, and Anders Søgaard. 2018. Sequence classification with human attention. In *Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL)*, pages 302–312, Brussels, Belgium.

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 (ACL)*, pages 579–584, Berlin, Germany.

Yevgeni Berzak, Chie Nakamura, Amelia Smith, Emily Weng, Boris Katz, Suzanne Flynn, and Roger Levy. 2022. CELER: A 365-participant corpus of eye movements in L1 and L2 English reading. *Open Mind*, pages 1–10.

Pieter Blignaut and Daniël Wium. 2014. Eye-tracking data quality as affected by ethnicity and experimental design. *Behavior Research Methods*, 46:67–80.

Lena Bolliger, David Reich, Patrick Haller, Deborah Jakobi, Paul Prasse, and Lena Jäger. 2023. ScanDL: A diffusion model for generating synthetic scanpaths on texts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 15513–15538, Singapore.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar.

Shuwen Deng, Paul Prasse, David Reich, Tobias Scheffer, and Lena Jäger. 2023a. Pre-trained language models augmented with synthetic scanpaths for natural language understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6500–6507, Singapore.

Shuwen Deng, David Reich, Paul Prasse, Patrick Haller, Tobias Scheffer, and Lena Jäger. 2023b. Eyettention: An attention-based dual-sequence model for predicting human scanpaths during reading. *Proceedings of the ACM on Human-Computer Interaction*, 7(ETRA):1–24.

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

Nora Hollenstein and Ce Zhang. 2019. Entity recognition at first sight: Improving NER with eye movement information. In *Proceedings of North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1–10, Minneapolis, Minnesota.

Lena Jäger, Silvia Makowski, Paul Prasse, Liehr Sascha, Maximilian Seidler, and Tobias Scheffer. 2020. Deep Eyedentification: Biometric identification using micro-movements of the eye. In *Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2019*, volume 11907 of *Lecture Notes in Computer Science*, pages 299–314, Cham, Switzerland.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy.

Marcel A Just and Patricia A Carpenter. 1980. A theory of reading: From eye fixations to comprehension. *Psychological Review*, 87(4):329.

Varun Khurana, Yaman Kumar, Nora Hollenstein, Rajesh Kumar, and Balaji Krishnamurthy. 2023. Synthesizing human gaze feedback for improved NLP performance. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 1895–1908, Dubrovnik, Croatia.

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*.

Shayne Longpre, Yu Wang, and Chris DuBois. 2020. How effective is task-agnostic data augmentation for pretrained transformers? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4401–4411, Online.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of International Conference on Learning Representations (ICLR)*, New Orleans, Louisiana, United States.

Silvia Makowski, Paul Prasse, David Reich, Daniel Krakowczyk, Lena Jäger, and Tobias Scheffer. 2021. Deepeyedentificationlive: Oculomotoric biometric identification and presentation-attack detection using deep neural networks. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(4):506–518.

Abhijit Mishra, Kuntal Dey, and Pushpak Bhattacharyya. 2017. Learning cognitive features from gaze data for sentiment and sarcasm classification using convolutional neural network. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 377–387, Vancouver, Canada.

Abhijit Mishra, Diptesh Kanojia, Seema Nagar, Kuntal Dey, and Pushpak Bhattacharyya. 2016. Leveraging cognitive features for sentiment analysis. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 156–166, Berlin, Germany.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, pages 8024–8035, Vancouver, Canada.

David Reich, Shuwen Deng, Marina Björnsdóttir, Lena Jäger, and Nora Hollenstein. 2024. Reading does not equal reading: Comparing, simulating and exploiting reading behavior across populations. In *Proceedings of the Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)*, pages 13586–13594, Turin, Italy.

Negar Sammaknejad, Hamidreza Pouretemad, Changiz Eslahchi, Alireza Salahirad, and Ashkan Alinejad. 2017. Gender classification based on eye movements: A processing effect during passive face viewing. *Advances in Cognitive Psychology*, 13(3):232.

Ekta Sood, Fabian Kögel, Philipp Müller, Dominike Thomas, Mihai Bâce, and Andreas Bulling. 2023. Multimodal integration of human-like attention in visual question answering. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 2648–2658, Vancouver, BC, Canada.

Ekta Sood, Simon Tannert, Philipp Müller, and Andreas Bulling. 2020. Improving natural language processing tasks with human gaze-guided neural attention. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*, pages 6327–6341, Online.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, pages 38–45, Online.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*, pages 6256–6268, Online.

Duo Yang and Nora Hollenstein. 2023. PLM-AS: Pretrained language models augmented with scanpaths for sentiment classification. In *Proceedings of the Northern Lights Deep Learning Workshop*, Tromsø, Norway.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2021. Revisiting few-sample BERT fine-tuning. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, Online.

A Model Details

Scanpath Generation Model For the utilization of the scanpath generation model Eyettention, we follow the work of (Deng et al., 2023a). The training process for the Eyettention model is conducted in two phases. First, we pre-train the Eyettention model on the L1 subset of the CELER corpus (Berzak et al., 2022), which comprises eye-tracking recordings collected from native speakers of English during natural reading sentences. Second, the Eyettention model is fine-tuned on downstream NLP tasks. More specifically, in our proposed scanpath-augmented fine-tuning scheme, we fine-tune the Transformer encoder and the Eyettention model, as well as train the scanpath encoder and the final dense layer from scratch. We tailor the parameters of Eyettention for specific downstream tasks, aiming to provide targeted inductive biases. For further details on the Eyettention model, please refer to (Deng et al., 2023b,a).

In our experiments, we evaluate our proposed approach using two distinct pre-trained LMs, BERT and RoBERTa, each equipped with its unique tokenizer. The Eyettention model includes a pre-trained LM in the text encoder for embedding the stimulus sentence. The generated fixation sequence (token index sequence) is based on the specific tokenizer associated with the pre-trained LM used. To facilitate a direct application of the arrangement operation based on the token-embedding sequence and fixation sequence without additional complex conversion, we maintain consistency by using the same pre-trained LMs in the Eyettention text encoder when evaluating specific pre-trained LMs as our base models. By replacing BERT with RoBERTa in the Eyettention text encoder, we observe a similar validation loss in scanpath prediction on the CELER corpus.

Scanpath Encoder The scanpath encoder is composed of a unidirectional GRU layer (Cho et al., 2014) with a hidden size of 768 and a dropout rate of 0.1. We initialize the hidden state of the GRU layer using the [CLS] token outputs from the final layer of the pre-trained LMs.

B Training Details

We train all models using the PyTorch (Paszke et al., 2019) library on an NVIDIA A100-SXM4-40GB GPU using the NVIDIA CUDA platform. We use the pre-trained checkpoints from the HuggingFace repository (Wolf et al., 2020) for the language model BERT$_{\text{base}}$ and RoBERTa$_{\text{base}}$. The models are optimized using the AdamW optimizer (Loshchilov and Hutter, 2019). We set the maximum sequence length to 128 and the training batch size to 32.

In the high-resource setting, we train the models for 20 epochs and update the best checkpoint by measuring validation accuracy every 500 steps. For datasets with fewer than 500 steps per epoch, we update and validate at the end of each epoch. We tune the learning rates for BERT from {5e-5, 4e-5, 3e-5, 2e-5} and for RoBERTa from {3e-5, 2e-5, 1e-5} for each task, following the recommendations in the original paper (Devlin et al., 2019; Liu et al., 2019).

In the low-resource setting, we train the models for 10 epochs and save checkpoints every epoch. We use the same learning rate that was found optimal in the high-resource setting for each task. We perform 5 runs with different data seeds ({111,222,333,444,555}) for shuffling, while the seed s=42 is consistently utilized for model training across all models.

In both high-resource and low-resource settings, for our proposed scanpath-augmented fine-tuning method, we conduct a hyperparameter search on the development set to determine the optimal trade-off factor λ for each task, exploring values from {1, 0.7, 0.5, 0.3, 0.1, 0.01, 0.001}. For the EDA baseline, we tune the number of generated augmented sentences added to the original training set, exploring values from {1, 2, 4, 8, 16} based on the recommendations in the original paper (Wei and Zou, 2019).