Synthesizing Human Gaze Feedback for Improved NLP Performance

Varun Khurana*
Adobe, IIIT Delhi
varun19124@iiitd.ac.in

Yaman Kumar Singla*
Adobe, IIIT Delhi, SUNY-Buffalo
ykumar@adobe.com

Nora Hollenstein
University of Copenhagen
nora.hollenstein@hum.ku.dk

Rajesh Kumar
Bucknell University
rajesh.kumar@bucknell.edu

Balaji Krishnamurthy
Adobe
kbalaji@adobe.com

Abstract

Integrating human feedback in models can improve the performance of natural language processing (NLP) models. Feedback can be either explicit (*e.g.* ranking used in training language models) or implicit (*e.g.* using human cognitive signals in the form of eyetracking). Prior eye tracking and NLP research reveal that cognitive processes, such as human scanpaths, gleaned from human gaze patterns aid in the understanding and performance of NLP models. However, the collection of *real* eyetracking data for NLP tasks is challenging due to the requirement of expensive and precise equipment coupled with privacy invasion issues. To address this challenge, we propose ScanTextGAN, a novel model for *generating* human scanpaths over text. We show that ScanTextGAN-generated scanpaths can approximate meaningful cognitive signals in human gaze patterns. We include synthetically generated scanpaths in four popular NLP tasks spanning six different datasets as proof of concept and show that the models augmented with generated scanpaths improve the performance of all downstream NLP tasks.

1 Introduction

Integrating human signals with deep learning models has been beginning to catch up in the last few years. Digital traces of human cognitive processing can provide valuable signals for Natural Language Processing (Klerke et al., 2016a; Plank, 2016). Various approaches for integrating human signals have been explored. For example, human feedback for better decisioning (Christiano et al., 2017), NLP tasks (Stiennon et al., 2020; Wu et al., 2021), and most recently language modeling using reinforcement learning with human feedback (RLHF) based reward (Bai et al., 2022; Ouyang et al., 2022). RLHF involves explicit human feedback and is expensive and hard to scale. On the other hand, previous studies have also tried to use implicit human feedback in the form of eyetracking signals. It has proven to be a useful signal for inferring human cognitive processing (Sood et al., 2020; Hollenstein and Zhang, 2019; Mathias et al., 2020). NLP researchers have focused on assessing the value of gaze information extracted from large, mostly dis-jointly labeled gaze datasets in recurrent neural network models (Ren and Xiong, 2021; Strzyz et al., 2019; Barrett et al., 2018a). The proposed approaches under this paradigm include gaze as an auxiliary task in multi-task learning (Klerke et al., 2016b; Hollenstein et al., 2019), as additional signals (Mishra et al., 2016b), as word embeddings (Barrett et al., 2018b), as type dictionaries (Barrett et al., 2016a; Hollenstein and Zhang, 2019), and as attention (Barrett et al., 2018a).

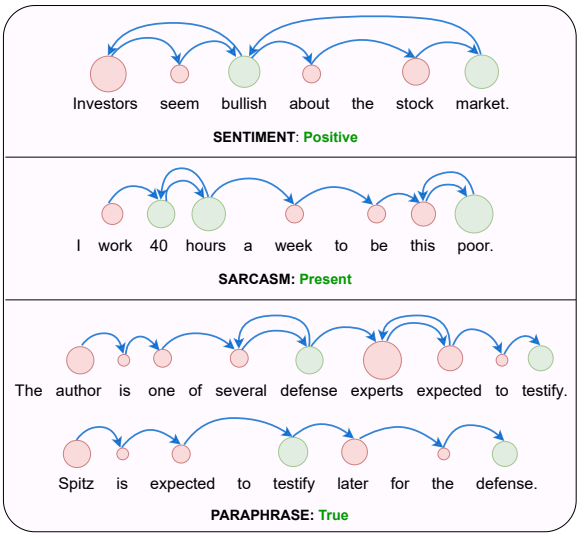

Figure 1: Generated scanpaths over text samples taken from various natural language processing (NLP) tasks. The green circles denote the important words characteristic of that task. The circles' size denotes the fixation duration, and the arrows depict the saccadic movements. As can be seen, linguistically important words often have a higher fixation duration and revisit. Regressions (word revisits) also appear in the examples.

Previous studies demonstrate that human scanpaths (temporal sequences of eye fixations, see Fig. 1) gleaned from eye tracking data improve the

* Equal Contribution

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 1895–1908

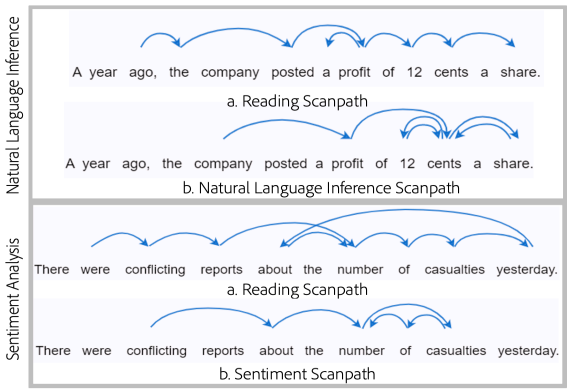

Figure 2: (Intent-aware) Scanpath samples generated by conditioning scanpath generation on different downstream natural language tasks. Note that the conditioned scanpaths are heavily biased to words important for that downstream task.

performance of NLP models. However, the real-world application of these methods remains limited primarily due to the cost of precise eye-tracking equipment, users' privacy concerns, and manual labor associated with such a setup. Therefore, generating scanpaths from existing eyetracking corpora would add great value to NLP research. To the best of our knowledge, this is the first paper to propose a model that generates scanpaths for a given read text with good accuracy. We call the model, ScanTextGAN.

We demonstrate the scanpath generation capability of ScanTextGAN over three eye-tracking datasets using multiple evaluation metrics. Further, we evaluate the utility of *generated* scanpaths for improvements in the performance of multiple NLP tasks (see Figs. 1,2) including the ones in the GLUE benchmark (Wang et al., 2018). The generated scanpaths achieve similar performance gains as the models trained with real scanpaths for classic NLP tasks like sentiment classification, paraphrase detection, entailment, and sarcasm detection.

Our contributions are threefold:

1. We propose ScanTextGAN, the first scanpath generator over text.

2. We compare ScanTextGAN with multiple baselines and conduct ablation experiments with varying models and configurations. The model performs well on the test sets and cross-domain generalization on two additional eyetracking datasets belonging to different text domains.

3. We tested the usefulness of generated scanpaths in downstream NLP tasks such as sentiment analysis, paraphrase detection, and sarcasm detection on six different datasets. The results show that the downstream NLP tasks benefited significantly from cognitive signals inherent in generated scanpaths. Further, we show how scanpaths change when finetuning with downstream natural language tasks (Figs.2,6) and that they lead to further improvements in downstream task performance (§4.3) showing how they can act as additional controls beyond the task architecture.

2 Related Work

When reading a text, humans do not focus on every word and often do not read sequentially (Just and Carpenter, 1980). A series of studies in psycholinguistics have shown that the number of fixations and the fixation duration on a word depend on several linguistic factors. The linguistic factors can also be determined given the cognitive features (Clifton Jr et al., 2007; Demberg and Keller, 2008). Though advances in ML architecture have helped bring machine comprehension closer to human performance, humans are still superior for most NLP tasks (Blohm et al., 2018; Xia et al., 2019).

It has been shown in the literature that integrating explicit (Bai et al., 2022; Ouyang et al., 2022) and implicit (cognitive processing) human feedback signals in traditional ML models is expected to improve their performance (Just and Carpenter, 1980). However, the cost of explicit feedback (e.g., using MTurk) and implicit feedback (e.g., eye tracking) at scale is excessively high. Similarly, privacy-invasive eye-tracking processes limit the scope of this idea. One way to address this problem is to use generated eye movements to unfold the full potential of eye-tracking research. Hence, the idea is to architect ScanTextGAN, a scanpath generator for text reading, and test its usefulness in downstream NLP tasks.

More precisely, this work builds upon previous works on 1) human attention modeling and 2) gaze integration in neural network architectures, which are described as follows:

Human Attention Modeling: Predicting what people visually attend to in images (saliency prediction) is a long-standing challenge in neuroscience and computer vision, the fields have seen many data-based models (Wang et al., 2021). In contrast to images, most attention models for eye movement behaviors during reading are cognitive process models, *i.e.*, models that do not involve machine learning but implement cognitive theories (Engbert et al., 2005; Xia et al., 2019). Key chal-

lenges for such models are a limited number of parameters and hand-crafted rules. Thus, it is difficult to adapt them to different tasks and domains and use them as part of end-to-end trained ML architectures (Kotseruba and Tsotsos, 2020). In contrast, learning-based attention models for text remain under-explored. Within that, all eye tracking models are saliency prediction models with non-existent work in predicting scanpaths. On the other hand, visual scanpaths generation for image-based eye tracking data has been recently explored for both traditional (Assens et al., 2019) and 360° images (Martin et al., 2022).

Matthies and Søgaard (2013) presented the first fixation prediction work for text. They built a person-independent model using a linear Conditional Random Fields (CRF) model. Hahn and Keller (2016) designed the Neural Attention Trade-off (NEAT) language model, which was trained with hard attention and assigned a cost to each fixation. Other approaches include sentence representation learning using surprisal and part of speech tags as proxies to human attention (Wang et al., 2017).

Our work differs from previous studies as we combine cognitive theory and data-driven approaches to predict scanpaths and further show its application in downstream NLP tasks (Hollenstein et al., 2021b,a).

Integrating Gaze in Network Architecture: Integration of human gaze data into neural network architectures has been explored for a range of computer vision tasks such as image captioning, visual question answering, and tagging (Karessli et al., 2017; Yu et al., 2017; He et al., 2019; Boyd et al., 2022). Hence, recent research has utilized features gleaned from readers' eye movement to improve the performance of complex NLP tasks such as sentiment analysis (Long et al., 2017; Mishra et al., 2016c), sarcasm detection (Mishra et al., 2016b), part-of-speech tagging (Barrett et al., 2016b), NER (Hollenstein and Zhang, 2019), and text difficulty (Reich et al., 2022).

While in recent years, eye tracking data has been used to improve and evaluate NLP models, the scope of related studies remains limited due to the requirement of real-time gaze data at inference time. Mathias et al. (2020) reported that there exists no automated way of generating scanpaths yet in the literature. With high-quality artificially generated scanpaths, the potential of leveraging eyetracking data for NLP can be unfolded. Additionally, generating scanpaths that mimic human reading behavior will help advance our understanding of the cognitive processes behind language understanding. Hence, we propose ScanTextGAN; researchers can use that to generate scanpaths over any text without worrying about collecting them from real users.

3 Proposed Model

In this section, we define the scanpath generation task, describe the ScanTextGAN model architecture, and provide details on loss functions and model training.

Task Definition: The task of scanpath generation is to generate a sequence $\mathcal{S}(\mathcal{T})$ representing a scanpath over the text $\mathcal{T} = \{w_1, w_2, ..., w_n\}$ composed of a sequence of words, can be defined as follows:

$$\mathcal{S}(\mathcal{T}) = \{.., (w_a^i, t^i), ..., (w_b^j, t^j), ..., (w_c^k, t^k)\} \quad (1)$$

where t^i represents the fixation duration over the word w_a occurring at the position i. Note that it is not necessary to have $a < b$ (words being read in linear order) or that $k = n$ (the number of fixations being equal to the number of words). Due to regressions, *i.e.*, backward saccades to previous words, words are also revisited. Hence, the same word could appear multiple times in the sequence.

3.1 ScanTextGAN Model Architecture

Fig. 3 illustrates the proposed conditional GAN architecture of the model. The ScanTextGAN model is composed of two competing agents. First, a conditional generator that generates scanpaths given text prompts. The second is a discriminator network, which distinguishes real human scanpaths from the generated ones. The ScanTextGAN model is trained by combining text content loss, scanpath content loss, and adversarial loss (Eq. 6). The scanpath content loss measures the difference between the predicted scanpath and the corresponding ground truth scanpath. The text content loss reconstructs the input text, and the adversarial loss depends on the real/synthetic prediction of the discriminator over the generated scanpath. We describe the losses along with the generator and discriminator architectures next.

Generator: The ScanTextGAN generator constitutes a transformer-based encoder-decoder framework. The encoder is conditioned on BERT-based text embeddings (Devlin et al., 2019), which are concatenated with noise to make the

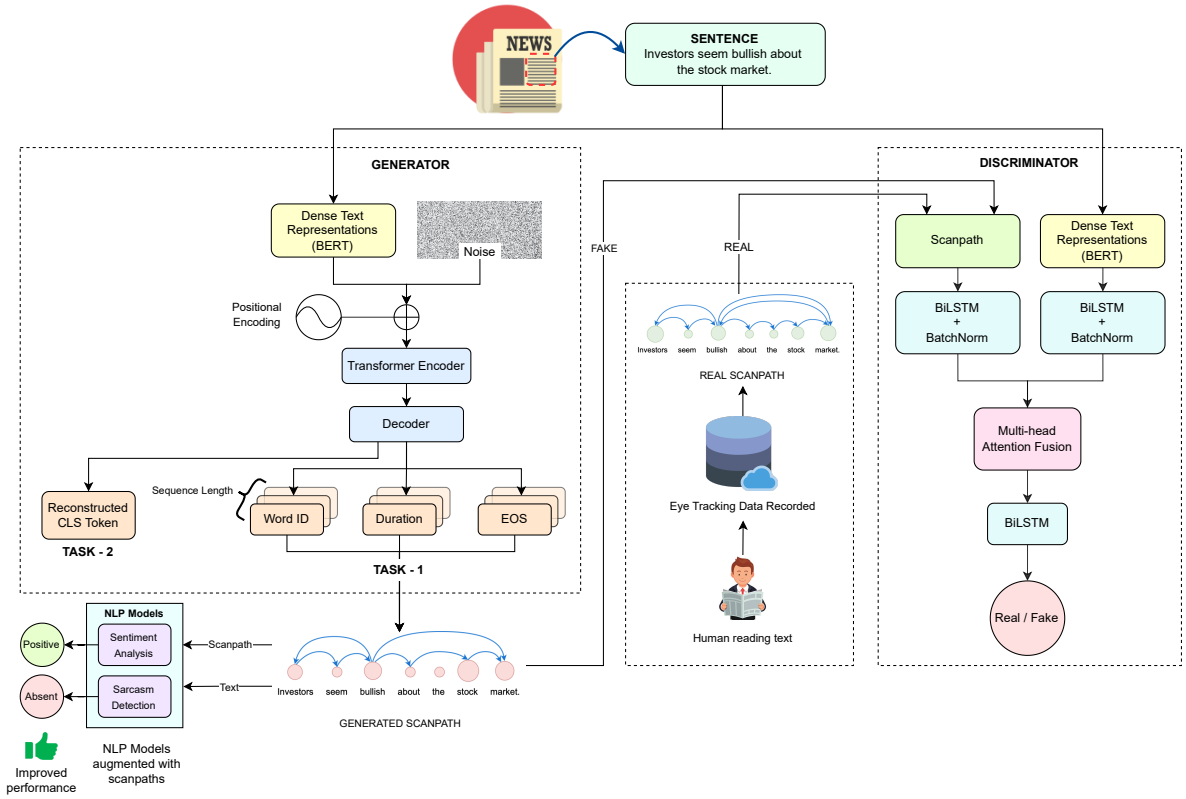

Figure 3: The architecture of the proposed **ScanTextGAN** model. The model consists of a conditional generator and a discriminator playing a zero-sum game. The generator is trained by two cognitively inspired losses: text content reconstruction and scanpath content reconstruction.

generator's output non-deterministic. The output of the transformer encoder is supplied to the decoder, which consists of task-specific feed-forward networks. One branch generates the scanpath (*Task 1*), while the other reconstructs the 768 dimensional CLS token embedding of the sentence (*Task 2*). The scanpath is output as a temporal sequence of word ID (fixation points) w_a^i, fixation duration t^i, and end-of-sequence probability EOS^i. At inference time, the length $L(G)$ of generated scanpath G is determined as follows:

$$L(G) = \begin{cases} \min_{1 \leq k \leq M}(k) & \text{if } EOS^k > \tau \\ M & \text{otherwise} \end{cases} \quad (2)$$

where M is the maximum scanpath length as described in section §3.2 and $\tau \in (0, 1)$ is a probability threshold. We use $\tau = 0.5$. The loss functions of the two branches are described below.

Scanpath Content Loss tries to minimize the deviation of generated scanpaths $\mathcal{G}(\mathcal{T}, \mathcal{N})$ from the ground-truth scanpaths $\mathcal{R}(\mathcal{T}, h))$ over text $\mathcal{T}$ where ground-truth scanpaths are recorded from the human h and $\mathcal{N}$ stands for Gaussian noise $\mathcal{N}(0, 1)$. The loss function $\mathbb{L}_s$ is given as:

$$\mathbb{L}_s(\mathcal{G}(\mathcal{T},\mathcal{N}), \mathcal{R}(\mathcal{T}, h)) = \frac{1}{k}\Sigma_{i=0}^{k}(\alpha(id_g^i - id_r^i)^2 + \beta(t_g^i - t_r^i)^2 + \gamma(E_g^i - E_r^i)^2) \quad (3)$$

which is a weighted sum of three terms. The first term measures the error between real and predicted *fixation points* given by the mean squared difference between generated and real word-ids $(id_g^i - id_r^i)$. It penalizes permutations of word ids and trains the model to approximate the real sequence of fixation points closely.

The second term measures the difference in *fixation durations* given by the mean squared difference between generated and real duration $(t_g^i - t_r^i)$. Fixation durations simulate human attention over words in the input text. Thus, a word with a larger fixation duration is typically synonymous with greater importance than other words in the input text. This error term supplements the generator's ability to learn human attention patterns over the input text.

Finally, the third term measures the mean squared error between the prediction of end-of-

sequence probability by real and generated distributions ($E_g^i - E_r^i$). These are weighted by the hyperparameters α, β, and γ. Preliminary experiments showed that optimizing the mean squared error leads to better performance over the cross-entropy loss for optimizing the EOS probability output.

Text Content Loss: Scanpaths depend heavily on the linguistic properties of the input text. Therefore, to guide the generator towards near the probable real data manifolds, we adopt reconstruction of the CLS token embedding of the input text (*Task 2*) by the generator as an auxiliary task since the CLS token embedding encodes a global representation of the input text. This text content reconstruction loss $\mathbb{L}_r$ is given as:

$$\mathbb{L}_r(\mathcal{G}(\mathcal{T},\mathcal{N}),\mathcal{R}(\mathcal{T},h)) = (BERT(w_i^g, w_j^g, ..., w_k^g - BERT(w_a^r, w_b^r, ...w_n^r))^2 \quad (4)$$

where $BERT(w_a^r, w_b^r, ...w_n^r)$ and $BERT(w_i^g, w_j^g, ...w_k^g)$ stand for the *CLS* vector representations of real and generated text respectively.

Discriminator: The goal of the discriminator is to distinguish between the real and synthetic scanpaths supplied to it. Similar to the generator, it requires text representations to distinguish between real and generated scanpaths. Specifically, the discriminator comprises two blocks of BiLSTMs that perform sequential modeling over the scanpaths and BERT embeddings. The outputs of the two branches are combined and passed to an attention fusion module with four heads, followed by another network of BiLSTMs. The hidden states of the last BiLSTM layer from both forward and backward directions are concatenated and supplied to a feed-forward network. A Sigmoid function activates the output of the feed-forward network. In this manner, the discriminator classifies the input scanpaths as either *real* or *fake*.

Adversarial Loss: The generator and discriminator networks are trained in a two-player zero-sum game fashion. The loss is given by:

$$\mathbb{L}_a = \min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x|\mathcal{T},h)] + \mathbb{E}_{z\sim p_z(z)}[1 - \log D(G(z|\mathcal{T},\mathcal{N}))] \quad (5)$$

Therefore, the net generator loss becomes:

$$\mathbb{L}_g = \mathbb{L}_s + \mathbb{L}_r + \mathbb{E}_{z\sim p_z(z)}[1 - \log D(G(z|\mathcal{T},\mathcal{N}))] \quad (6)$$

3.2 Dataset

For training the ScanTextGAN model, we use the CELER dataset (Berzak et al., 2022). It contains eyetracking data of 365 participants for nearly 28.5 thousand newswire sentences, sourced from the Wall Street Journal Penn Treebank (Marcinkiewicz, 1994). Each participant in CELER reads 156 newswire sentences. Half of the sentences are shared across participants, and the rest is unique to each participant. The maximum sentence length was set to 100 characters. Participant eyetracking data were recorded using Eyelink 1000 tracker in a desktop mount configuration with a sampling rate of 1000 Hz. The ScanTextGAN model is trained to approximate the average eye movements of all the participants who read given sentences. The CELER dataset was envisioned to enable research on language processing and acquisition and to facilitate interactions between psycholinguistics and natural language processing. Furthering the goal, we use it to train our conditional GAN model through which we show human scanpath approximation capabilities (§4.2). Also, we use it to show improvements in the performance of NLP tasks (§4.3).

The data consist of tuples of participant ID, sentence ID, and word ID corresponding to fixation point and fixation duration. We compute the 99th percentile of fixation durations and treat it as the largest value. Fixations of durations longer than this are treated as outliers and hence dropped from the dataset. To apply the scanpath reconstruction loss (Eq. 3), we scale all fixation durations by the maximum value and then normalize them to [0,1]. Similarly, word IDs in each sentence are normalized to [0, 1] after scaling them by the length of that sentence. For the last fixation point in every scanpath, the binary EOS token is set to 1. The maximum scanpath length is set to 80 fixation points (99th percentile of the lengths). Thus shorter scanpaths are padded while longer scanpaths are trimmed. We use BERT to encode the sentences and obtain their 768-dimensional embeddings, keeping the max length parameter as 80, thus resulting in an 80×768 dimensional tensor.

3.3 Parameter Settings

Sinusoidal positional encoding is applied over the input embeddings fed to the generator. We use a 3-layer transformer encoder with four head attention and a hidden dimension size of 776 in the generator. In the discriminator, we use

bidirectional LSTMs over sentence embeddings and generated scanpaths with a hidden size of 64 and a dropout ratio of 0.3, followed by batch normalization for faster convergence. An attention module with four attention heads is applied after concatenating the outputs. We employ the Adam and RMSProp optimizer to minimize generator and discriminator losses. The batch size is set to 128, the initial learning rate of the generator to 0.0001, and that of the discriminator to 0.00001. The model is trained for 300 epochs. Our implementation uses PyTorch, a popular deep-learning framework in Python. All experiments are run on an Intel Xeon CPU with Nvidia A100-SXM GPUs.

4 Performance Evaluation

We quantify the performance of ScanTextGAN in two regimes[1]; first, scanpath generation with three datasets, and second, NLP tasks with six datasets. Similar to prior computer vision studies (Sun et al., 2019; de Belen et al., 2022; Kümmerer and Bethge, 2021; Jiang et al., 2016), we evaluate the ScanTextGAN model over the scanpath generation task. For this, we use the test split of the CELER dataset, Mishra et al. (2016a), and Mishra et al. (2017). In addition, unlike the computer vision studies, we also evaluate the ScanTextGAN model for improvement in NLP tasks. The hypothesis is that the human eyes (and consequently the brain) process many language comprehension tasks unconsciously and without visible effort. The next logical step is to capture (or, in our case, generate) this mental representation of language understanding and use it to improve our machine-learning systems. For evaluation, we use four tasks from the GLUE benchmark and two from the tasks proposed by Mishra et al. (2016a). While the ScanTextGAN model is trained over news text from the CELER dataset, with the help of the other datasets, we expand our testing to other domains, including reviews, quotes, tweets, and Wikipedia text.

4.1 Evaluation Datasets

Mishra et al. (2017) comprises eye movements and reading difficulty data recorded for 32 paragraphs on 16 different topics, *viz.* history, science, literature, *etc*. For each topic, comparable paragraphs were extracted from Wikipedia[2] and simple Wikipedia[3]. The participant's eye movements are tracked using an SR-Research Eyelink-1000 Plus eye tracker. Using the ground truth scanpaths over the text corpora, we evaluate the quality of generated scanpaths.

Mishra et al. (2016a) contains eye fixation sequences of seven participants for 994 text snippets annotated for sentiment and sarcasm. These were taken from Amazon Movie Corpus , Twitter, and sarcastic quote websites. The task assigned to the participants was to read one sentence at a time and annotate it with binary sentiment polarity labels (*i.e.*, positive/negative). The same datasets were used in several studies (Joshi et al., 2015; Mishra et al., 2016b,c) to show improvements in sarcasm and sentiment analysis. We use the datasets to evaluate both the generation quality and potential improvements in NLP tasks.

Furthermore, we explore the potential of including cognitive signals contained in scanpaths in NLP models for a range of GLUE tasks which include Sentiment Analysis using Stanford Sentiment Treebank (SST), Paraphrase Detection using Microsoft Research Paraphrase Corpus (MRPC) and Quora Question Pairs (QQP), Natural Language Inference using Recognizing Textual Entailment (RTE) dataset.

Next, we cover the results of scanpath generation and its application in NLP tasks.

4.2 Evaluation of Scanpath Generation

We evaluate the scanpath generation model on two most commonly used metrics in image scanpath generation studies (Sun et al., 2019; Chen and Sun, 2018; de Belen et al., 2022; Kümmerer et al., 2022): **MultiMatch** (Jarodzka et al., 2010) and **Levenshtein Distance** (Levenshtein, 1965). Multimatch is a geometrical measure that compares scanpaths across a comprehensive set of dimensions composed of shape, lengths, position, and fixation duration. Levenshtein Distance between a pair of sequences measures the least number of edits (inserts, deletes, substitution) to transform one into the other. More details are discussed in Appendix:A.

Further, as a top-line comparison, we use **inter-subject scanpath similarity** (Sun et al., 2019). It

[1]All results are calculated with five random seeds and reported as the mean of those five runs

[2]https://en.wikipedia.org/

[3]https://simple.wikipedia.org/

[4]In the CELER dataset, there are only 78 shared sentences amongst all the participants. Therefore, inter-subject scanpath evaluation is done only for these sentences. In contrast, the ScanTextGAN results are reported for the entire test set (including these 78 sentences).

Generator Model	MultiMatch ↑				Levenshtein Distance ↓
	Vector↑	Length↑	Position↑	Duration↑	
Inter-subject score[4]	0.973	0.958	0.830	0.698	0.691
LSTM Encoder-Decoder trained with scanpath content loss	0.975	0.956	0.765	0.344	0.865
ScanTextGAN – Text Reconstruction – GAN Loss	0.968	0.947	0.728	0.703	0.779
ScanTextGAN	**0.983**	**0.972**	**0.787**	0.733	**0.769**
ScanTextGAN – Text Reconstruction	0.974	0.957	0.773	0.703	0.798
ScanTextGAN – GAN Loss	0.973	0.955	0.750	**0.761**	0.786
ScanTextGAN + addition of noise	0.971	0.952	0.756	0.736	0.791
ScanTextGAN – Text (CLS) Reconstruction + sentence reconstruction	0.978	0.963	0.724	0.721	0.805

Table 1: In-domain Evaluation of Scanpath Generation on the CELER dataset (Berzak et al., 2022).

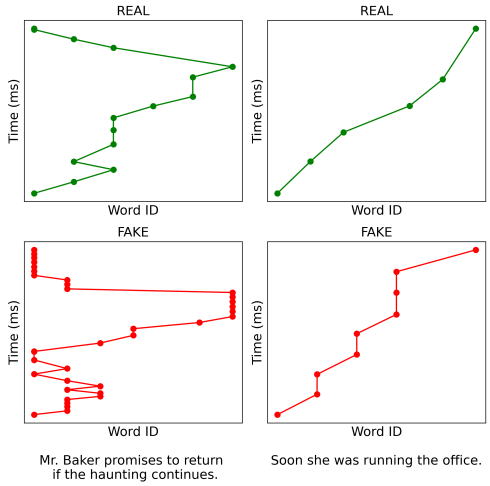

Figure 4: Comparison of *real* and *synthesized* scanpaths corresponding to a few text samples. The proposed ScanTextGAN model generates the latter.

measures the degree of variation among real human scanpaths corresponding to each text input. To compute this, we first calculate each subject's performance by treating the scanpaths of other subjects as the ground truth. Then, the average value of all subjects is used as inter-subject performance.

Baselines: Since ScanTextGAN is the first text-based scanpath generation model, we conduct an ablation study to compare ScanTextGAN with its other variants. Specifically, we compare ScanTextGAN with the following six configurations: (1) An LSTM-based network trained with scanpath content loss. Sentence embeddings obtained through BERT are concatenated with noise in this model. The resultant is fed to an attention module with four heads, then passed to a network of LSTMs and Batch Normalization layers applied in tandem. (2) ScanTextGAN model trained with only the scanpath content loss. (3) ScanTextGAN model without the text reconstruction loss (Task-2). (4) ScanTextGAN model with BERT-based sentence embeddings reconstruction instead of CLS token reconstruction. (5) ScanTextGAN model with the addition of noise instead of concatenation. (6) ScanTextGAN model trained without GAN loss.

Results: Table 1 presents the results of our scanpath prediction model on the CELER dataset. Further, we also compare ScanTextGAN with baselines on two other contemporary datasets of movie reviews, tweets, and sarcastic quotes (Mishra et al., 2016a), Wikipedia and simple Wikipedia paragraphs (Mishra et al., 2017). Tables 2 and 3 present the results of our model on those datasets. For obtaining results on these corpora, we use the model trained on the CELER dataset, thus helping us evaluate the cross-domain performance of the model.

As can be seen in Table 1, Table 2 and Table 3, ScanTextGAN outperforms other models for scanpath prediction on most metrics. The performance of ScanTextGAN even surpasses inter-subject reference on Duration and comes very close to Vector, Length, and Position.

We observe that adopting the reconstruction of the CLS token as an auxiliary task (Task - 2) boosts the model performance. Reconstructing the full sentence embeddings rather than the CLS tokens only as an auxiliary task does not always improve the results, despite adding a larger computational overhead. The results also reveal that concatenating noise with text embeddings is more rewarding than adding it.

Further, to compare the skipping behavior of ScanTextGAN with humans, we calculate the weighted F1 score of the words skipped and attended by both model types. We find the weighted F1 to be 64.6 between them. Fig. 4 presents a visual comparison between real scanpaths from the available eyetracking data and scanpaths generated by ScanTextGAN, corresponding to some randomly chosen text samples. We can observe that the generated scanpaths resemble the real ones to a great

Generator Model	MultiMatch ↑				Levenshtein Distance ↓
	Vector↑	Length↑	Position↑	Duration↑	
Inter-subject score	0.977	0.963	0.839	0.715	0.723
LSTM Encoder-Decoder trained with scanpath content loss	**0.984**	**0.973**	0.714	0.379	0.918
ScanTextGAN – Text Reconstruction – GAN Loss	0.977	0.960	0.780	0.769	0.847
ScanTextGAN	0.966	0.945	**0.791**	**0.771**	**0.836**
ScanTextGAN – Text Reconstruction	0.976	0.961	0.763	0.757	0.845
ScanTextGAN – GAN Loss	0.976	0.959	0.774	0.768	0.839
ScanTextGAN + addition of noise	0.968	0.947	0.737	0.743	0.838
ScanTextGAN – Text (CLS) Reconstruction + sentence reconstruction	0.964	0.934	0.747	0.733	0.869

Table 2: Cross-domain Evaluation of Scanpath Generation on the Dataset by Mishra et al. (2016a).

Generator Model	MultiMatch ↑				Levenshtein Distance ↓
	Vector↑	Length↑	Position↑	Duration↑	
Inter-subject score	0.994	0.991	0.834	0.620	0.845
LSTM Encoder-Decoder trained with scanpath content loss	**0.992**	**0.987**	0.596	0.329	0.969
ScanTextGAN – Text Reconstruction – GAN Loss	0.990	0.984	0.729	0.705	0.951
ScanTextGAN	0.984	0.977	**0.759**	0.693	**0.931**
ScanTextGAN – Text Reconstruction	0.986	0.981	0.756	**0.706**	0.939
ScanTextGAN – GAN Loss	0.990	0.984	0.739	**0.706**	0.945
ScanTextGAN + addition of noise	0.984	0.976	0.759	0.703	0.943
ScanTextGAN – Text (CLS) Reconstruction + sentence reconstruction	0.983	0.974	0.667	0.674	0.958

Table 3: Cross-domain Evaluation of Scanpath Generation on the Dataset by Mishra et al. (2017).

extent. Thus, the quantitative and qualitative results on in-domain and cross-domain settings lead us to believe that our proposed scanpath generation model can be deemed a good approximator of the human scanpaths.

4.3 Application to NLP Tasks

We use them to augment various NLP models and measure their performance to demonstrate the usefulness of cognitive signals hidden in the *generated* scanpaths.

Sentiment Classification and Sarcasm Detection: For these tasks, we use a model consisting of a network of two branches of BiLSTMs and Batch Normalization layers that perform sequential modeling over text representations obtained through BERT and scanpaths fed as input to the model. The outputs of both branches are combined and passed to another layer of BiLSTMs, followed by a feed-forward network that predicts binary sentiment/sarcasm labels corresponding to the input after activating with the Sigmoid function. We follow a 10-fold cross-validation regime.

We compare the models with generated scanpaths, real scanpaths, and without scanpaths. Further, to investigate whether performance gains observed by adding scanpaths are due to scanpaths and not the increase in the number of parameters, we train a *Random-Random* variant in which we send Random noise as scanpaths to the model with an increased number of parameters. We also simulate the real-world case where both real and generated scanpaths are available during train time, but only generated ones are available during test time, for example, during user deployment.

Model Configuration		F1 score	
Train	**Test**	**Sentiment**	**Sarcasm**
w/o	w/o	0.7839	0.9438
Random	Random	0.7990	0.9397
Random	Generated	0.7773	0.9313
Real	Generated	0.8319	0.9378
Real	Real	0.8334	0.9501
Generated	Real	0.8402	0.9452
Generated	Generated	0.8332	0.9506
Real + Generated	Generated	**0.8404**	**0.9512**
Intent-Aware	Intent-Aware	**0.8477**	**0.9528**

Table 4: Sentiment analysis and sarcasm detection results on the dataset by Mishra et al. (2016a). Model configuration refers to the type of scanpath included in train and test data.

Table 4 records the results of sentiment analysis and sarcasm detection tasks (Mishra et al., 2016a). We note that generated scanpaths training and testing lead to similar gains for sentiment analysis and sarcasm detection as real scanpaths. The model with an increased number of parameters fed random noise in place of scanpaths performs similarly to the model trained without any scanpaths. Interestingly, the best results are obtained when model training uses both real and generated scanpaths. We believe this is due to ScanTextGAN bringing additional cognitive information from the news-reading CELER corpus, which is not present in the

Dataset	Model	Acc	F1 score
SST	w/o scanpaths	0.8090	0.8089
	w/ random scanpaths	0.8059	0.8061
	w/ generated scanpaths	**0.8138**	**0.8138**
	w/ intent-aware scanpaths	**0.8269**	**0.8272**
MRPC	w/o scanpaths	0.6902	0.6656
	w/ random scanpaths	0.6623	0.6680
	w/ generated scanpaths	**0.6969**	**0.6828**
	w/ intent-aware scanpaths	**0.7009**	**0.6911**
RTE	w/o scanpaths	0.6162	0.6080
	w/ random scanpaths	0.5802	0.5794
	w/ generated scanpaths	**0.6211**	**0.6205**
	w/ intent-aware scanpaths	**0.6293**	**0.6278**
QQP	w/o scanpaths	0.8499	0.8513
	w/ random scanpaths	0.8491	0.8503
	w/ generated scanpaths	**0.8578**	**0.8596**
	w/ intent-aware scanpaths	**0.8648**	**0.8658**

Table 5: Results of training NLP models with and without scanpaths on the GLUE benchmark tasks. Including scanpaths leads to consistent improvements across all the NLP tasks.

real scanpaths in Mishra et al. (2016a). In addition to the intrinsic evaluation presented in §4.2, this downstream evaluation demonstrates the high quality of the synthesized scanpaths, showing that they contain valuable cognitive processing signals for NLP tasks.

GLUE Tasks: To validate further, we augment classification models (based on sequential modeling using LSTMs) with generated scanpaths to show performance improvement in downstream NLP tasks on four GLUE benchmark datasets – SST, MRPC, RTE, QQP as described in §4.1. Table 5 reports the accuracy and weighted-F1 scores of the models trained with and without scanpaths for these tasks. We observe that in all four tasks, the model trained with generated scanpaths outperforms the one without scanpaths.

Intent-Aware Scanpaths: Finally, we try to condition scanpaths generation on the downstream natural language task. We back-propagate gradients from the downstream NLP task to the conditional generator. In this fashion, the model learns to generate *intent-aware* scanpaths. The hypothesis is that finetuning scanpath generation based on feedback from the natural language task will bias the generator towards words more pertinent to that task and thus could help further improve performance on the downstream task. The architecture is shown in Appendix: Fig 5. The results in Tables 4 and 5 validate the hypothesis that we observe consistent improvements in all downstream tasks. Fig 2 and Appendix: Fig 6 show a few examples of scanpaths and saliency generated for three downstream natural language tasks.

Together these results corroborate the hypothesis that leveraging the cognitive signals approximated by synthetic scanpaths in NLP models leads to performance gains.

5 Conclusion

In this work, we make two novel contributions toward integrating cognitive and natural language processing. (1) We introduce the first scanpath generation model over text, integrating a cognitive reading model with a data-driven approach to address the scarcity of human gaze data on text. (2) We propose generated scanpaths that can be flexibly adapted to different NLP tasks without needing task-specific ground truth human gaze data. We show that both advances significantly improve performance across six NLP datasets over various baselines. Our findings demonstrate the feasibility and significant potential of combining cognitive and data-driven models for NLP tasks. Without the need for real-time gaze recordings, the potential research avenues for augmenting and understanding NLP models through the cognitive processing information encoded in synthesized scanpaths are multiplied.

6 Limitations

In this work, we demonstrated artificial scanpath generation over multiple eye-tracking datasets. Further, our experiments build a link between cognitive and natural language processing and show how one can inform the other. However, the proposed method has a few limitations, which we aim to address in the future. The field needs work on bigger and more diverse eye-tracking datasets, which can enable scanpath generation over longer text sequences and can model generating scanpaths conditioned on previously read context. Besides, a better understanding of the entire scanpath generation process can help model the intra and inter-sentence scanpath generation process. The understanding would enable the integration of scanpaths to generative modeling tasks, which we intend to take up in future work. Another parallel direction is to include both explicit (like using RLHF) and implicit signals (like using cognitive signals) to better NLP tasks like language modeling.

References

Marc Assens, Xavier Giro-i Nieto, Kevin McGuinness, and Noel E. O'Connor. 2019. Pathgan: Visual scanpath prediction with generative adversarial networks. In *Computer Vision – ECCV 2018 Workshops*. Springer International Publishing.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

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

Maria Barrett, Joachim Bingel, Frank Keller, and Anders Søgaard. 2016a. 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, Berlin, Germany. Association for Computational Linguistics.

Maria Barrett, Ana Valeria González-Garduño, Lea Frermann, and Anders Søgaard. 2018b. Unsupervised induction of linguistic categories with records of reading, speaking, and writing. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2028–2038, New Orleans, Louisiana. Association for Computational Linguistics.

Maria Barrett, Frank Keller, and Anders Søgaard. 2016b. Cross-lingual transfer of correlations between parts of speech and gaze features. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1330–1339, Osaka, Japan. The COLING 2016 Organizing Committee.

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*.

Matthias Blohm, Glorianna Jagfeld, Ekta Sood, Xiang Yu, and Ngoc Thang Vu. 2018. Comparing attention-based convolutional and recurrent neural networks: Success and limitations in machine reading comprehension. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 108–118, Brussels, Belgium. Association for Computational Linguistics.

Aidan Boyd, Kevin W Bowyer, and Adam Czajka. 2022. Human-aided saliency maps improve generalization of deep learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2735–2744.

Zhenzhong Chen and Wanjie Sun. 2018. Scanpath prediction for visual attention using ior-roi lstm. In *IJCAI*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Charles Clifton Jr, Adrian Staub, and Keith Rayner. 2007. Eye movements in reading words and sentences. *Eye movements*.

Filipe Cristino, Sebastiaan Mathôt, Jan Theeuwes, and Iain D Gilchrist. 2010. Scanmatch: A novel method for comparing fixation sequences. *Behavior research methods*.

Ryan Anthony Jalova de Belen, Tomasz Bednarz, and Arcot Sowmya. 2022. Scanpathnet: A recurrent mixture density network for scanpath prediction. In *IEEE CVPR*.

Vera Demberg and Frank Keller. 2008. Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Ralf Engbert, Antje Nuthmann, Eike M Richter, and Reinhold Kliegl. 2005. Swift: a dynamical model of saccade generation during reading. *Psychological review*.

Michael Hahn and Frank Keller. 2016. Modeling human reading with neural attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 85–95, Austin, Texas. Association for Computational Linguistics.

Sen He, Hamed R Tavakoli, Ali Borji, and Nicolas Pugeault. 2019. Human attention in image captioning: Dataset and analysis. In *ICCV*.

Nora Hollenstein, Maria Barrett, Marius Troendle, Francesco Bigiolli, Nicolas Langer, and Ce Zhang. 2019. Advancing nlp with cognitive language processing signals. *arXiv preprint arXiv:1904.02682*.

Nora Hollenstein, Emmanuele Chersoni, Cassandra L. Jacobs, Yohei Oseki, Laurent Prévot, and Enrico Santus. 2021a. CMCL 2021 shared task on eye-tracking prediction. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 72–78, Online. Association for Computational Linguistics.

Nora Hollenstein, Federico Pirovano, Ce Zhang, Lena Jäger, and Lisa Beinborn. 2021b. Multilingual language models predict human reading behavior. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 106–123, Online. Association for Computational Linguistics.

Nora Hollenstein and Ce Zhang. 2019. Entity recognition at first sight: Improving NER with eye movement information. 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 1–10, Minneapolis, Minnesota. Association for Computational Linguistics.

Halszka Jarodzka, Kenneth Holmqvist, and Marcus Nyström. 2010. A vector-based, multidimensional scanpath similarity measure. In *Proceedings of the 2010 symposium on eye-tracking research & applications*.

Ming Jiang, Xavier Boix, Gemma Roig, Juan Xu, Luc Van Gool, and Qi Zhao. 2016. Learning to predict sequences of human visual fixations. *IEEE transactions on neural networks and learning systems*.

Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 757–762, Beijing, China. Association for Computational Linguistics.

Marcel Just and Patricia A. Carpenter. 1980. A theory of reading: From eye fixations to comprehension.

Nour Karessli, Zeynep Akata, Bernt Schiele, and Andreas Bulling. 2017. Gaze embeddings for zero-shot image classification. In *IEEE CVPR*.

Sigrid Klerke, Yoav Goldberg, and Anders Søgaard. 2016a. Improving sentence compression by learning to predict gaze. In *NAACL: Human Language Technologies*, San Diego, California. Association for Computational Linguistics.

Sigrid Klerke, Yoav Goldberg, and Anders Søgaard. 2016b. Improving sentence compression by learning to predict gaze. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1528–1533, San Diego, California. Association for Computational Linguistics.

Iuliia Kotseruba and John K Tsotsos. 2020. 40 years of cognitive architectures: core cognitive abilities and practical applications. *Artificial Intelligence Review*.

Matthias Kümmerer and Matthias Bethge. 2021. State-of-the-art in human scanpath prediction. *arXiv preprint arXiv:2102.12239*.

Matthias Kümmerer, Matthias Bethge, and Thomas SA Wallis. 2022. Deepgaze iii: Modeling free-viewing human scanpaths with deep learning. *Journal of Vision*.

V Levenshtein. 1965. Leveinshtein distance.

Yunfei Long, Qin Lu, Rong Xiang, Minglei Li, and Chu-Ren Huang. 2017. A cognition based attention model for sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 462–471, Copenhagen, Denmark. Association for Computational Linguistics.

Mary Ann Marcinkiewicz. 1994. Building a large annotated corpus of english: The penn treebank. *Using Large Corpora*.

Daniel Martin, Ana Serrano, Alexander W. Bergman, Gordon Wetzstein, and Belen Masia. 2022. Scangan360: A generative model of realistic scanpaths for 360° images. *IEEE Transactions on Visualization and Computer Graphics*.

Sandeep Mathias, Diptesh Kanojia, Abhijit Mishra, and Pushpak Bhattacharya. 2020. A survey on using gaze behaviour for natural language processing. In *IJCAI*. Survey track.

Franz Matthies and Anders Søgaard. 2013. With blinkers on: Robust prediction of eye movements across readers. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 803–807, Seattle, Washington, USA. Association for Computational Linguistics.

Abhijit Mishra, Diptesh Kanojia, and Pushpak Bhattacharyya. 2016a. Predicting readers' sarcasm understandability by modeling gaze behavior. In *AAAI*.

Abhijit Mishra, Diptesh Kanojia, Seema Nagar, Kuntal Dey, and Pushpak Bhattacharyya. 2016b. Harnessing cognitive features for sarcasm detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1095–1104, Berlin, Germany. Association for Computational Linguistics.

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

Abhijit Mishra, Diptesh Kanojia, Seema Nagar, Kuntal Dey, and Pushpak Bhattacharyya. 2017. Scanpath complexity: Modeling reading effort using gaze information. *AAAI*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al.

2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Barbara Plank. 2016. Keystroke dynamics as signal for shallow syntactic parsing. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 609–619, Osaka, Japan. The COLING 2016 Organizing Committee.

David Robert Reich, Paul Prasse, Chiara Tschirner, Patrick Haller, Frank Goldhammer, and Lena A. Jäger. 2022. Inferring native and non-native human reading comprehension and subjective text difficulty from scanpaths in reading. In *ETRA*. Association for Computing Machinery.

Yuqi Ren and Deyi Xiong. 2021. CogAlign: Learning to align textual neural representations to cognitive language processing signals. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3758–3769, Online. Association for Computational Linguistics.

Ekta Sood, Simon Tannert, Philipp Müller, and Andreas Bulling. 2020. Improving natural language processing tasks with human gaze-guided neural attention. *NeurIPS*.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

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, Hong Kong, China. Association for Computational Linguistics.

Wanjie Sun, Zhenzhong Chen, and Feng Wu. 2019. Visual scanpath prediction using ior-roi recurrent mixture density network. *IEEE transactions on pattern analysis and machine intelligence*, 43(6):2101–2118.

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. Association for Computational Linguistics.

Shaonan Wang, Jiajun Zhang, and Chengqing Zong. 2017. Learning sentence representation with guidance of human attention. In *IJCAI*. AAAI Press.

Wenguan Wang, Qiuxia Lai, Huazhu Fu, Jianbing Shen, Haibin Ling, and Ruigang Yang. 2021. Salient object detection in the deep learning era: An in-depth survey. *IEEE T-PAMI*.

Jeff Wu, Long Ouyang, Daniel M Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. 2021. Recursively summarizing books with human feedback. *arXiv preprint arXiv:2109.10862*.

Menglin Xia, Ekaterina Kochmar, and Ted Briscoe. 2019. Automatic learner summary assessment for reading comprehension. 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 2532–2542, Minneapolis, Minnesota. Association for Computational Linguistics.

Youngjae Yu, Jongwook Choi, Yeonhwa Kim, Kyung Yoo, Sang-Hun Lee, and Gunhee Kim. 2017. Supervising neural attention models for video captioning by human gaze data. In *IEEE CVPR*.

A Scanpath Evaluation Metrics

MultiMatch is a geometrical measure that models scanpaths as vectors in 2-D space, wherein the vectors represent saccadic eye movements. Starting and ending coordinates of these saccades constitute the fixation positions. It compares scanpaths across multiple dimensions, *viz.* shape, length, position, direction, and fixation duration. Shape measures the vector difference between aligned saccade pairs, which is then normalized by twice the diagonal screen size. Length measures the normalized difference between the endpoints of real and generated saccade vectors. Direction is the angular distance between the two vectors. The position is the Euclidean difference in position between aligned vectors, and duration measures the difference in fixation durations normalized against the maximum duration. Since our work deals with scanpaths over text, we use 1-D space to represent the saccade vectors where word IDs denote the fixation positions. Thus, it is easy to see that computing scanpath direction similarity is redundant here (it is subsumed within position); hence we drop it from our analysis.

Levenshtein Distance between a pair of sequences measures the least number of character edits, i.e., insertion, deletion, and substitution needed to transform one sequence into the other. Specifically, we use it to gauge the degree of dissimilarity between a pair of real R and generated G scanpaths. To account for the fixation durations of each word, R and G are temporally binned using a 50 ms bin size, similar to the computation of ScanMatch metric (Cristino et al., 2010). The resulting sequences of word IDs, R_W and G_W are transformed into character strings, $R_S = \{r_1, r_2, ..., r_n\}$ and $G_S = \{g_1, g_2, ..., g_m\}$, where R_S and G_S are strings over the ASCII alphabet and $n = |R_S|$ and $m = |G_S|$.

Levenshtein Distance (LD) between strings R_S and G_S is computed and then normalized by the length of the longer string, which yields a Normalized Levenshtein Distance (NLD) score, as given below:

$$NLD = \frac{LD(G_S, R_S)}{\max(|R_S|, |G_S|)} \tag{7}$$

Thus, a lower NLD score is indicative of greater scanpath similarity.

B Intent-Aware Scanpaths

As described in section §4.3, the generator conditioned on the downstream natural language task yields *intent-aware* scanpaths. Augmenting NLP models with these scanpaths leads to higher performance gains. Here, we provide more details on *intent-aware* scanpath generation. Please refer to figures 5 and 6 on the following page. Saliency corresponding to intent-aware scanpaths are shown in Fig. 6.

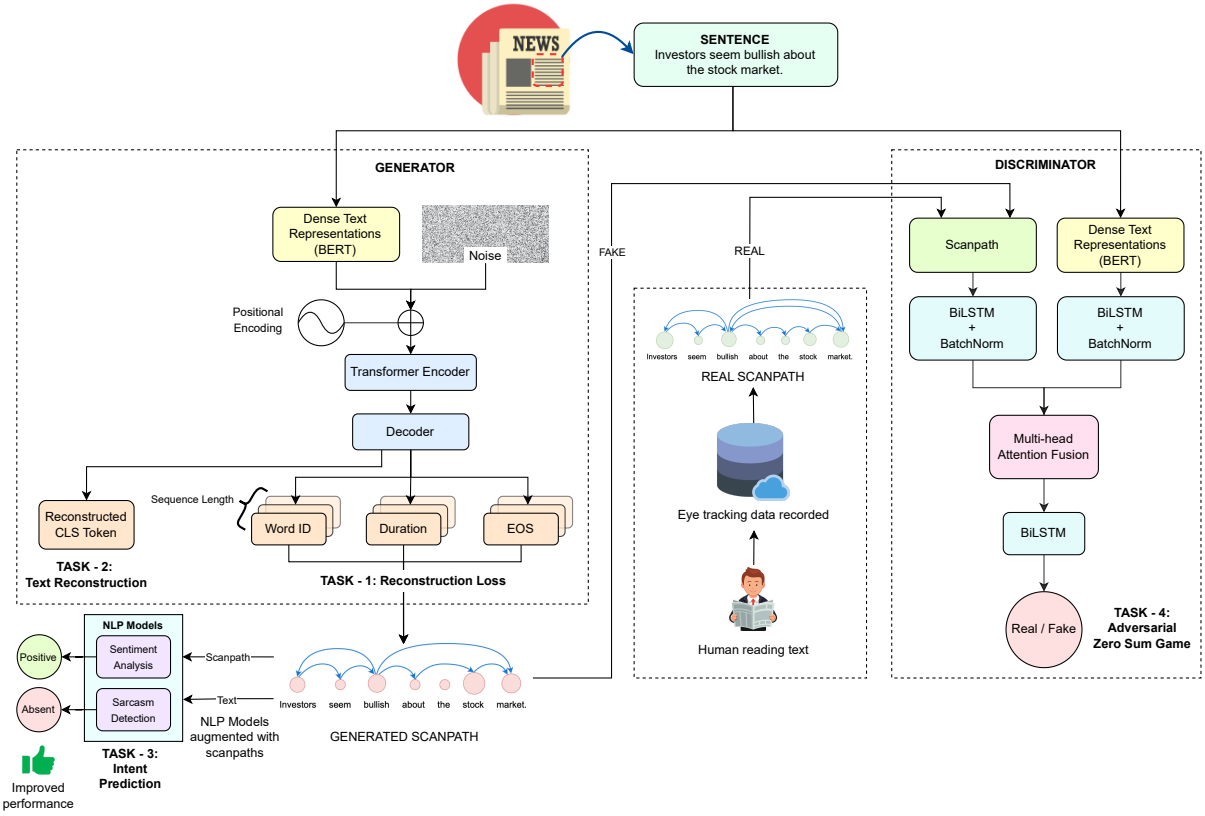

Figure 5: The architecture of the proposed Intent-Aware **ScanTextGAN** model. The model consists of a conditional generator and a discriminator playing a zero-sum game. Two cognitively inspired losses train the generator: scanpath (Task-1) and text (Task-2) reconstruction, a loss from the downstream intent of the natural language task (Task-3), and finally, the loss from the adversarial zero-sum game (Task-4). Variations of scanpaths are generated based on the downstream natural language task.

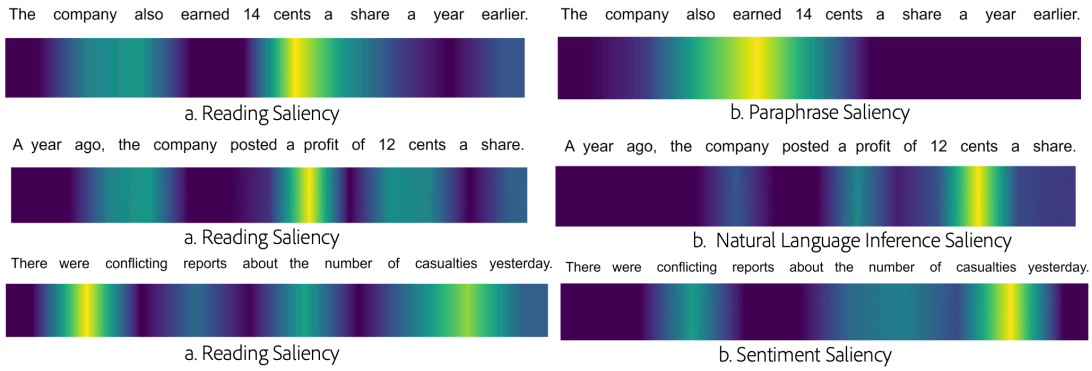

Figure 6: Saliency samples generated by conditioning scanpath generation on different downstream natural language tasks. It can be observed that the conditioned saliency pays much more attention to words important for that downstream task.