

GazBy: Gaze-Based BERT Model to Incorporate Human Attention in Neural Information Retrieval

Sibo Dong*

sd1242@georgetown.edu

InfoSense, Dept. of Computer Science
Georgetown University, USA

Justin Goldstein

jjg130@georgetown.edu

InfoSense, Dept. of Computer Science
Georgetown University, USA

Grace Hui Yang

grace.yang@georgetown.edu

InfoSense, Dept. of Computer Science
Georgetown University, USA

ABSTRACT

This paper is interested in investigating whether human gaze signals can be leveraged to improve state-of-the-art search engine performance and how to incorporate this new input signal marked by human attention into existing neural retrieval models. In this paper, we propose GazBy (**G**aze-based **B**ert model for document relevancy), a light-weight joint model that integrates human gaze fixation estimation into transformer models to predict document relevance, incorporating more nuanced information about cognitive processing into information retrieval (IR). We evaluate our model on the Text Retrieval Conference (TREC) Deep Learning (DL) 2019 and 2020 Tracks. Our experiments show encouraging results and illustrate the effective and ineffective entry points for using human gaze to help with transformer-based neural retrievers. With the rise of virtual reality (VR) and augmented reality (AR), human gaze data will become more available. We hope this work serves as a first step exploring using gaze signals in modern neural search engines.

CCS CONCEPTS

• Information systems → Question answering; Learning to rank.

KEYWORDS

Neural information retrieval, Gaze, Human attention, BERT, Transformer

ACM Reference Format:

Sibo Dong, Justin Goldstein, and Grace Hui Yang. 2022. GazBy: Gaze-Based BERT Model to Incorporate Human Attention in Neural Information Retrieval. In *Proceedings of the 2022 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '22)*, July 11–12, 2022, Madrid, Spain. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3539813.3545129>

1 INTRODUCTION

Neural retrievers apply deep neural networks to solving information retrieval problems such as document retrieval and passage retrieval. Neural networks are given a task to score a query-document pair based on the relevance of the document to the query. These

methods are usually categorized into *representation-based* (e.g., DSSM [26] and SNRM [74]) and *interaction-based* neural retrievers (e.g., DRMM [18], KRNM [10], and DeepTileBars [62]).

Representation-based neural retrievers encode a query q and a document d separately and interact them later when calculating a relevance score between q and d . Each query and each document is encoded into a single fixed-size embedding vector. On the other hand, interaction-based neural retrievers generate a query-document interaction matrix in indexing-time or in query-time and feed it into neural networks to predict a relevance score between q and d . Note that in representation-based methods, a query-document interaction appear late at the retrieval stage; whereas in interaction-based methods, the query-document interaction appears early since they take care of the (q, d) pair as a whole starting from the very beginning – either from the indexing phase (e.g. when building an inverted index as in conventional retrieval methods) or from the beginning of a re-ranking process.

Currently, nearly all the most effective neural retrieval models, including MonoBERT [45] and ColBERT [33], make use of BERT [5, 6, 11, 38], a deep language model (LM) pre-trained by the transformer architecture [64] on the tasks of Masked Language Modeling and Next Sentence Prediction. MonoBERT [45] is an example of interaction-based neural retriever directly using the BERT embeddings. It inputs a '[CLS]' token appended to the front of a query-document pair as a whole input sequence to a series of encoder layers for all-in-all interactions and scores the query-document relevance based on the learned representation of the [CLS] token. On the other hand, ColBERT [33] is an example of representation-based neural retriever directly using the BERT embeddings. It creates query and document representations separately and passes them to two encoders and proposes *MaxSim*, a weighted summation of cosine similarity scores of the most similar document term to each query term, to calculate the query-document relevance. They are also known as *cross-encoder* and *bi-encoder* transformer-based models, respectively, due to the difference in their neural network architectures. In this paper, we will frequently refer to their architectures as cross-encoder and bi-encoder, respectively.¹

BERT embeddings have also been widely used in derived functions by recent neural retrievers, mostly to obtain meaningful term-level weights. For instance, DeepCT [9] learns a linear aggregation function on BERT embeddings $a \cdot E_w(d) + b$ via regression, where $E_w(d)$ is a word w 's BERT embedding in document d . It can be thought of an aggregated contextual term weight for w in d .

¹Note that the word 'interaction' in "interaction-based retriever" and the word 'interaction' in "query-document interaction" and "late interaction" have different meanings. The former refers to a retriever's setting or architecture, where the latter refers to an actual operation between a query and a document. To avoid confusion, we choose to use cross-encoder and bi-encoder to call the two types of retrievers, instead of calling them "interaction-based" and "representation-based" as in most IR papers.

*Equal contribution between the first two authors. Both are first authors to the paper.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICTIR '22, July 11–12, 2022, Madrid, Spain

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-9412-3/22/07...\$15.00

<https://doi.org/10.1145/3539813.3545129>

EPIC [39] used a max operation on the softplus function over BERT embeddings to select the most similar term in a document. DeepImpact [40] learned a Multi-layer Perceptron on BERT embeddings as term weight; and TILDE [78] obtained a term’s log conditional probability $\log P_\theta(w|d)$ by applying a language modeling head on the [CLS] token of word w ’s BERT embedding in d . Adopting the BERT representation, either directly using it or deriving from it, in neural retrievers have become a major research interest in Information Retrieval (IR).

However, the natural language processing (NLP) community reflects that looking inside of BERT’s transformer blocks and layers, its attention head weights are sometimes questionable. Abnar and Zuidema [1] have shown that the BERT’s attention head weights “often approximate an almost uniform distribution in higher model layers.” Hollenstein and Beinborn showed that BERT’s attention head weights correlate poorly with human language processing patterns [22]. Gao and Callan [16] also pointed out that “BERT directly out of pre-training has a non-optimal attention structure” and they proposed to short-circuit BERT’s information flow across different transformer layers in a dense retriever. All these suggest that despite the unprecedented success these BERT-based neural retrievers have achieved, the core element that they all leverage, the vanilla BERT, is not as perfect as we think.

One recent attempt to improve the vanilla BERT embeddings is to seek help from another source of attention, *human gaze*. Human gaze modeling used to be dominated by models that implement cognitive theories with hand-crafted features (e.g., the E-Z Reader [54], Uber Reader [65], and the SWIFT Model [13]), which, although easy to explain, are often difficult to use in machine learning pipelines [61]. Modern human gaze modeling uses neural networks to predict gaze fixation durations. Popular methods include the trade-off model [20], gradient-based method [22], Recurrent Neural Networks (RNNs), and word skipping probability prediction [20, 42].

The use of human gaze can be found in computer vision [30, 59, 70, 72, 73] and natural language processing (NLP) [34, 57, 61, 76] tasks, such as visual question answering [51], object referral [63], paraphrasing and sentence compression [61], sequence labeling [34], classifying pronouns [71], key phrase extraction [76], and prediction of multi-word expressions [57]. Human gaze incorporates more nuanced information about human cognitive processing because it is highly correlated with relative importance for reading comprehension [41, 53]. The use of gaze has been explored in the IR literature, too [3, 19, 35, 37, 58]. It was mostly used as a way to predict text salience and to understand relevance. However, existing work does not target pre-trained transformer models for ad hoc retrieval. To the best of our knowledge, human gaze attention has not been applied to document or passage retrieval in neural retrievers.

In this paper, we explore incorporating gaze signals into two of the most effective transformer-based neural information retrieval methods [38]. One for MonoBERT [46], and another for ColBERT [33]. With a light-weighted human gaze prediction model, we test one proof-of-concept that if we can enhance the computational attention scores in transformers using human attention scores and improve ad hoc retrieval. We propose GazBy (**G**aze-based **B**ert model for document relevancy), a joint model that integrates human gaze

fixation estimation into transformer models to predict document relevance, incorporating more nuanced information about cognitive processing into IR. Our joint model has a gaze prediction component and a relevance scoring component. We train our gaze prediction model using the GECO [4] and Zuco [24] gaze datasets and test it with both MonoBERT and ColBERT on the passage re-ranking task in the Text Retrieval Conference (TREC) Deep Learning (DL) 2019 and 2020 Tracks [5, 6].

A major effort in this work is that we investigate *how* to incorporate human gaze attention into transformer-based retrieval models. We explore extensively in this work to find out the best ways to integrate human gaze attention with computational contextual attention to help with ad hoc retrieval.

Our finding is that it is important to incorporate human gaze attention at a specific moment during the entire retrieval pipeline. Our experiments show that the **human attention should be merged into a transformer-based retrieval model when its query-document interaction happens**:

- For MonoBERT, where its interaction operation happens early during the all-in-all attention interaction, especially when the interaction is close to finish, we find it is best to merge the human gaze predictions with the last attention layer;
- For ColBERT, where its interaction operation happens late during the calculation of *MaxSim*, we find it is best to merge the human gaze predictions into the retriever during this calculation.

We also investigate other places to merge gaze in their neural network architectures, however, they produce significantly poorer results. For instance, if we incorporate the gaze attention scores into the attention mechanism in ColBERT’s transformer layer, which happens before the interaction, it hurts the system’s effectiveness significantly. Why does where the query-document interaction happen matter? We think that the influential weighting for each token representation by gaze fixation prediction should happen during the query and document interaction because gaze prediction itself is another form of interaction between query and document tokens. This interaction occurs through our gaze prediction model’s transformer layers. By weighting the interaction between the query and document tokens through gaze-based human attention we align interaction operations among our gaze prediction model and our pre-trained transformer models, thus allowing our model to produce promising results in the cases described above.

To summarize, our paper makes the following contributions:

- (1) A mechanism for incorporating human gaze attention into transformer-based neural retrieval models;
- (2) A proof-of-concept of this mechanism on two state-of-the-art transformer-based neural information retrieval models that show promising results;
- (3) An investigation of where and how to merge human gaze prediction in a neural information retrieval model.

2 RELATED WORK

2.1 Neural Information Retrieval

Ad hoc retrieval aims to find which are the best documents D for a query q , structurally it is a task that involves two inputs, the query

q and the document d . It must perform an interaction, also known as Cartesian mathematically, between the two inputs and yields a single relevance score for the pair. This unavoidable interaction appear in different forms in different retrieval algorithms. We can find it as a similarity scoring function $score(q, d)$ (e.g., cosine similarity, dot product, neural matching kernels, and *MaxSim* in ColBERT) in traditional retrieval models and in the interaction-based neural retrieval methods; a joint operation between q and d (e.g., when building the inverted index in traditional retrieval methods, and the neural interaction in interaction-based neural retrieval methods); and the “scaled dot-product attention” [64] in the transformer layers of the transformer-based neural retrievers.

Neural Information Retrieval is the application of deep neural networks to solving information retrieval problems such as document retrieval and passage retrieval. The methods can be categorized into representation-based (e.g., DSSM [26] and SNRM [74]) and interaction-based (e.g., DRMM [18], KRNM [10], and DeepTileBars [62]).

Representation-based retrievers use a bi-encoder setting, which encode a query q and a document d separately and interact them later when calculating a similarity score between them. Each query and each document is encoded into a single fixed-size embedding vector. The embedding vectors can be either dense or sparse, which correspond to the recently popular dense retrievers (e.g., DPR[31], SBERT [55], Condenser [16], ICT [36], RocketQA [52], ANCE [69], and RepBERT[75]) and sparse retrievers (e.g. SparTerm [2], SPLADE [15], and EPIC [39]). They study pre-training or fine-tuning (mostly fine-tuned from the BERT embeddings [11]) methods to obtain low-dimensional encodings for query and for documents. Unlike results obtained from BoW representations, top-K results obtained from these learned representations cannot be efficiently found without any approximation [7, 29, 69]. ColBERT [33] is a type of representation-based retriever.

Interaction-based retrievers [12, 14, 18, 47–49, 56, 62, 68] generate a query-document interaction matrix in indexing-time or query-time and feed it into neural networks to predict a relevance score. Interaction-based retrievers (both pre-neural and neural) have been known for their better retrieval effectiveness. For instance, the very successful BM25 method is a sparse, interaction-based method in the pre-neural era. Early interaction-based neural retrieval models are not efficient and can only be used for re-ranking the results generated from a first-stage retriever. More recently, efforts have been made to create indices for these neural retrieval methods to save query-time calculations and support them to directly perform first-stage, full-length document retrieval [9, 74]. MonoBERT [45] uses a cross-encoder setting and is type of interaction-based retrieval. It employs an all-in-all fashion of extensive interaction among query terms and document terms.

In this paper, we find *when the interaction happens* a crucial piece of information to determine where to merge human gaze attention into transformer-based retrieval models. Query-document interaction can appear early in an algorithm, such as during indexing time when building the inverted index in traditional retrieval methods, at the early joint operation in interaction-based neural retrieval models, and during the all-to-all interaction in MonoBERT. It can

also appear late in an algorithm. For instance, the last step of similarity calculation in representation-based neural retrieval methods, and the *MaxSim* operator in ColBERT.

2.2 Gaze and Human Attention

Our work is inspired by the use of human gaze and the improvement gained by it in computer vision [30, 59, 70, 72, 73] and natural language processing (NLP) [34, 57, 61, 76] tasks. Qiao et al. used gaze as an optimization objective for visual question answering [51] and for filtering out irrelevant information, as in the case of using gaze for object referral in videos [63]. Henderson et al. demonstrated that video viewers without given an explicit task are likely trying to understand the meaning of a scene and direct their gaze accordingly [21]. Sood et al. 2020 used gaze to inform the tasks of paraphrasing and sentence compression [61]. Others in the field of NLP have also successfully used gaze for sequence labeling [34], classifying pronouns [71], key phrase extraction [76], and prediction of multi-word expressions [57].

Human gaze modeling used to be dominated by models that implement cognitive theories with hand-crafted features, which, although easy to explain, are often difficult to use in machine learning pipelines [61]. Popular models include the E-Z Reader [54], Uber Reader [65], and the SWIFT Model [13]. Modern modeling methods use neural networks to predict human sentence processing. For instance, Keller and Mahn used a Neural Attention Trade-off language model (NEAT) [20] to optimize the number of gaze fixations as a trade-off between information and one’s time and energy. Hollenstein et al. have trained BERT to predict gaze in a multilingual corpus [23]. Keller [32] and Michaelov and Bergen [43] used Recurrent Neural Networks (RNNs) to predict gaze and indicated that RNN’s architecture appears suited to the task. Matthies and Sogaard [42] and Hahn and Keller [20] have used neural models to predict word skipping probability, which is directly related to gaze fixation duration. These uses and modeling of gaze have been found to improve machine learning model performance on human language-related tasks compared to using models without gaze input or supervision.

2.3 Use of Gaze in IR and Neural IR

The use of gaze has been explored in IR literature before the deep learning era. In 2003, Salojärvi et al. [58] first studied eye movements and IR to learn whether it is possible to determine relevance from eye movements. Balatsoukas and Ruthven [3] explored the relationship between the use of correlation criteria and the level of correlation (relevant, partially relevant, irrelevant) based on a collection of eye movements, such as the number and length of fixations. Gwizdka and Zhang [19] examined correlations from the perspective of users’ pupil dilation during web page visits and revisits and demonstrated the feasibility of predicting the relevance of web documents from eye-tracking data. Li et al. [37] performed extensive user study on eye-movement and user attention distribution during relevance-oriented reading comprehensions and created a prediction model to infer human attention during relevance judgments. Lagun and Agichtein segmented web pages and develop a generative model, *MICS* [35], to predict text salience. They modeled user attention based on the amount of time a user views a

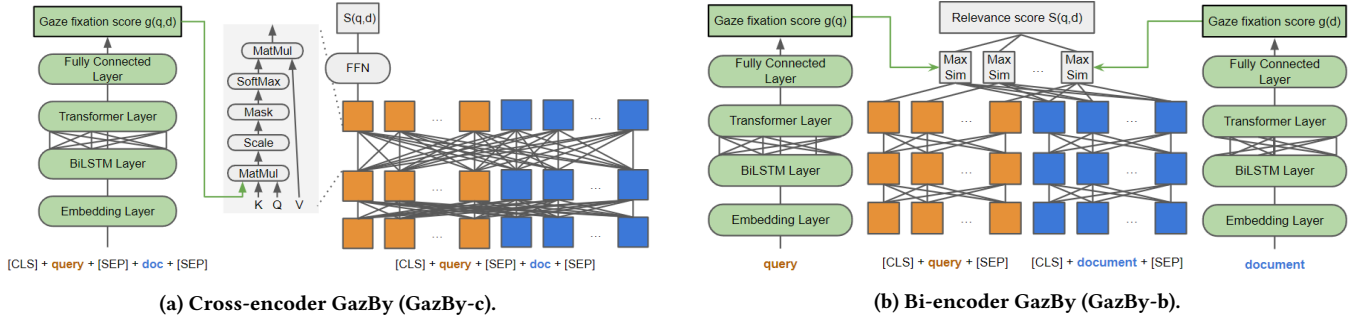


Figure 1: Architecture of GazBy.

web page segment on their browser along with features that describe the content of that web page segment. They demonstrated improvements on the web search, news, Wikipedia, shopping, and social network web page domains for retrieving relevant search results. Their methods, however, does not target neural information retrieval.

To the best of our knowledge, human gaze attention has not been applied to document or passage retrieval in neural retrieval models.

3 GAZBY: GAZE-BASED BERT MODEL FOR DOCUMENT RELEVANCY

In this paper, we propose GazBy, a light-weight joint model that combines human gaze fixation estimation into transformer-based retrieval models, with the aim to introduce more nuanced information about a text into information retrieval. We explore our idea with both MonoBERT and ColBERT, where MonoBERT is a cross-encoder (and interaction-based retriever) that performs early interactions between the query and document and ColBERT is a bi-encoder (and representation-based retriever) that performs the interaction late in its retrieval process.

3.1 Architecture

Figure 1 shows the overall architecture of GazBy. GazBy has two main components. The first is a gaze fixation prediction model. The second component is a transformer-based retrieval model. Depending on the architecture of the transformer-based retriever, GazBy combines the gaze prediction model differently with it. Cross-encoders are transformer-based retrieval models that create embeddings for q - d pairs using a single encoder, while bi-encoders embed q and d separately [27]. They correspond to interaction-based and representation-based neural retrievers, respectively. We therefore have two variations of GazBy: GazBy-c designed for cross-encoders (interaction-based) such as MonoBERT and GazBy-b for bi-encoders (representation-based) such as ColBERT.

The green parts of Figure 1 show the architecture of our gaze prediction model. According to [61] and [66], the combination of a BiLSTM [17] with a subsequent transformer network has the most similar predictions to human gaze processing patterns. Our gaze prediction model used in GazBy-c and GazBy-b is identical. It first takes in tokens in a pre-processed text, which could be a query or a document (as in GazBy-b), or the concatenation of a query and a

document (as in GazBy-c). The tokens are fed into an embedding layer, followed by a BiLSTM layer with 128 hidden units and four self-attention transformer layers with four attention heads. Finally a fully connected feedforward network (FFN) is used to predict a gaze fixation score $g(x_i)$ for each token x_i in text \vec{x} :

$$\vec{x} \xrightarrow{\text{BiLSTM, Transformer, FFN}} g(\vec{x}) \quad (1)$$

The higher the predicted gaze score, the longer the gaze fixation duration over a token.

Figure 1a illustrates the proposed cross-encoder GazBy (GazBy-c). It concatenates the query and the document as one input, and feeds the input into the gaze prediction model to obtain gaze fixation scores for the words. These gaze fixation scores are then used to modify a cross-encoder retrieval model's (e.g., MonoBERT) encoder layers. It is done by weighting the attention scores by the gaze scores when the encoder performs multi-head scaled dot product attention. The body of our cross-encoder adopts the BERT Large architecture, using 24 attention layers with 16 attention heads each. This joint cross-encoder model judges the relevance by the learned [CLS] token using a feed forward neural network:

$$\begin{aligned} \text{Concat}(q, d) &\xrightarrow{\text{BiLSTM, Transformer, FFN}} g(q, d) && \text{gaze prediction} \\ \text{Concat}(q, d) &\xrightarrow{L-1 \text{ Encoder Layers}} E_{L-1}^{(q, d)} && \text{input embeddings} \\ K_i &:= E_{L-1} \times W^{K_i} \\ V_i &:= E_{L-1} \times W^{V_i} \\ Q_i &:= E_{L-1} \times W^{Q_i} \\ E_L &= \text{Softmax}\left(\frac{Q \times (K \odot G)^T}{\sqrt{\dim}}\right) \times V && \text{modified attention} \\ E_L^{[\text{CLS}]} &\xrightarrow{\text{FFNS}(q, d)} && \text{relevance score using [CLS]} \end{aligned}$$

where K_i, V_i, Q_i are *Key, Value, Query* representations for multi-head scaled dot product attention of attention head i . Note this query is not our query, but an internal variable in an attention layer. In addition, \dim is the second dimension of *Key, Value, Query* in each head. G is $g(q, d)$ repeated over $n = q_{len} + d_{len}$ rows and E_L is embeddings obtained after the l^{th} encoder layer. $W^{K_i}, W^{V_i}, W^{Q_i}$ are the tensors of learned parameters that map E_{L-1} to $K_i, V_i, Q_i \in n \times \dim$ for the i^{th} attention head in the l^{th} encoder layer.

Table 1: Statistics of Gaze datasets.

	GECO [4]	ZuCo [24]
# words	54,364	20,293
# sentences	5,300	1,049
Avg. sentence Length	10.64 (std=8.20)	19.36 (std=9.26)

Figure 1b is the bi-encoder (e.g., ColBERT) variant of GazBy (GazBy-b). It sends the query and the document as two separate inputs to the gaze prediction model to obtain their gaze fixation scores accordingly. It also sends the query and the document separately into a BERT Large encoder, which performs multi-head scaled dot product attention over the query or the document terms alone. Eventually, we calculate a (q,d) relevance score using a modified *MaxSim* function, which is a weighted summation of cosine similarities of the most similar document term to each query term, by weighting each similarity using its gaze fixation score:

$$\begin{aligned}
 q &\xrightarrow{\text{BiLSTM, Transformer, FFN}} g(q) && \text{gaze prediction} \\
 d &\xrightarrow{\text{BiLSTM, Transformer, FFN}} g(d) \\
 q &\xrightarrow{\text{BERT Large}} E_L^q && \text{query embeddings} \\
 d &\xrightarrow{\text{BERT Large}} E_L^d && \text{document embeddings} \\
 S(q, d) &= \text{MaxSim}(E_L^q \odot g(q), E_L^d \odot g(d)) && \text{relevance score}
 \end{aligned}$$

Next, we describe components of GazBy-b and GazBy-c in details.

3.2 Gaze Fixation Prediction

Gaze fixations are valuable signals from a human user that directly showcase the amount of attention the user spends while reading and processing a piece of text [53]. Given an information need, gaze fixation can indicate which words in a document and how much they capture a user’s attention according to the need. In this work, we present a gaze fixation prediction model that aims to estimate the fixation duration as a score $g(x)$ for each token x_i in a string of text $x = [x_1, x_2, \dots, x_n]$. The bigger the value of $g(x_i)$, the longer the gaze fixation period at token x_i . [61] and [66] found that the combination of a BiLSTM with a subsequent transformer network has the most similar predictions to human gaze processing patterns, with four layers and four attention heads each specially. Below we detail our gaze model’s design following their finding.

An input text sequence is first tokenized and then padded with special tokens. The Wordpiece [67] tokenizer is used to split input string x into a sequence of tokens x_1, x_2, \dots, x_n . To be compatible with the transformer-based retrieval models, we add special tokens such as [CLS] and [SEP] into the input embeddings, following [45]. [CLS] is used in classification tasks and usually at the very beginning of the text; [SEP] is used to separate two input text sequences. Some words are split into smaller subwords, e.g., ‘wifi’ to [‘wi’, ‘##fi’] and ‘bluetooth’ to [‘blue’, ‘##tooth’], which helps decompose the out-of-vocabulary words or unknown tokens and retain some partial meaning. We also pad the tokens with another special token [PAD] to keep all inputs the same length (empirically

set to 10 in our implementation) before feeding them into the neural network. The input text is thus turned into [CLS], x_1, x_2, \dots, x_n , [SEP], [PAD], ..., [PAD].

The neural network for gaze prediction starts with an embedding layer. GloVe embedding [50] is used to encode each token into a 300-dimension vector. We then feed the entire embedding into a single BiLSTM layer and pass it onto to a four-layers four-head self-attention transformer layer. Finally, we feed the resulting embeddings into a fully connected layer to get the gaze fixation prediction of each input token. Note that special tokens, such as [CLS], [SEP], and [PAD], are non-existing words and should not have any fixations on them. We thus label all the special tokens as [ZERO] in their ground truth labels during training, which prevents the gaze prediction model from scoring them and flattening the weights for meaningful words. Subwords are labeled with the same ground truth gaze fixation label as the word it is a part of.

Our gaze fixation prediction model is trained on two English eye-tracking datasets Ghent Eye Tracking Corpus (GECO) [4] and Zurich Cognitive Language Processing Corpus (ZuCo) [24, 25]. The GECO dataset collected eye-tracking data when native English speakers reading the entire novel *The Mysterious Affair at Styles*, which was presented in paragraphs to them on a screen. The ZuCo corpus [24] collected eye-tracking records from native English speakers when they read movie reviews, biographical sentences, and English Wikipedia. Table 1 shows the statistics of the two datasets. They are combined into a single collection and standardize their gaze fixation target value into the range of [0, 1]. As we can see, they are quite small, summing into a total of around 6,000 sentences. The neural network’s training and optimization are done using Adam with a learning rate of 0.0001. We trained the model for 100 epochs and achieved a 10-fold cross-validation mean squared error of 0.004.

3.3 Cross-Encoder GazBy (GazBy-c)

This subsection demonstrates how our joint model GazBy-c incorporates human gaze fixations into MonoBERT, a state-of-the-art cross-encoder transformer-based retrieval model.

The proposed process takes the following steps. First, we tokenize each query-document pair and insert [CLS] and [SEP] tokens: [CLS] + $x_1 \dots x_{q_{len}}$ + [SEP] + $x_1 \dots x_{d_{len}}$ + [SEP]. We input this concatenated q-d pair into the gaze fixation prediction model.

Second, we also feed this q-d pair into $L - 1$ encoder layers out of L total, where each encoder layer uses multi-head attention, adding and normalization, and a feed forward network, identical to Vaswani et al.’s encoder layers [64]. The input sequence to the first encoder layer is the Wordpiece embeddings of the input token sequence. Each subsequent encoder layer receives the output of the previous encoder layer as input. Multi-head attention represents the input embeddings of each encoder layer in a set of three tensors for each of several attention heads, a key tensor, K_i , a value tensor, V_i , and a query tensor, Q_i , using matrix multiplication against weight matrices W^{K_i} , W^{V_i} , and W^{Q_i} respectively, for the i^{th} attention head. The attention that occurs in the first $L-1$ encoder layers takes the form: $\text{Softmax}(\frac{Q \times K^T}{\sqrt{dim}}) \times V$. The “attention” scores here are formed by the matrix $Q \times K^T$, which denote the amount of attention that token i pays to token j within a sequence of length n .

Third, we perform a modified multi-head attention in the last encoder layer. It is done such that each token’s attention scores to all other tokens are weighted by the predicted fixation duration of the token being attended to. In scaled dot product attention [64], this equates to multiplying the key layer by each term’s fixation duration, expanded to size dim , the number of columns in the key layer representation of the query-document sequence. The newly proposed attention mechanism is

$$Softmax\left(\frac{Q \times (K \odot G)^T}{\sqrt{dim}}\right) \times V, \quad (2)$$

where we denote element-wise multiplication by \odot and matrix multiplication by \times . This scaled attention mechanism ensures that the most important terms have a greater impact on a query-document pair score. Words receiving more gaze have a greater impact because the attention matrix in the embedding space of $n \times n$ is meant to demonstrate how much each word in row i attends to each word in column j . By element-wise multiplying the K tensor by gaze $G \in n \times dim$, we scale the i^{th} embedding’s attention to the j^{th} embedding by the j^{th} embedding’s gaze score when K and Q tensors interact in the attention tensor $Q \times K^T$.

In addition to this setting for GazBy-c, we also attempt to incorporate gaze weighted scaled dot product attention as described above to all layers in the encoder (GazBy-c All Layers), as well as by multiplying the Wordpiece embeddings input into the first layer using the gaze scores obtained in the first step above (GazBy-c First Layer). For results and further description, see Section 4.

Finally, we train the parameters of the gaze fixation prediction model and encoder layers’ parameters by minimizing the cross-entropy loss on relevance labels as a binary classification problem.

$$Loss = - \sum_{k \in R^+} \log(S(q, d)_k) - \sum_{k \in R^-} \log(1 - S(q, d)_k), \quad (3)$$

where R^+ are the relevant query and document pairs, and R^- are the irrelevant pairs in our training data. $S(q, d)_k$ is the score for that (q,d) pair by GazBy-c.

3.4 Bi-Encoder GazBy (GazBy-b)

This subsection presents our gaze-based joint model, GazBy-b, for bi-encoder transformer-based retrieval models such as ColBERT [33]. Bi-encoder retrieval models feed the query and the document into two encoders and learn their representations separately and then the query-document interaction happens when calculating the (q,d) relevance score.

In this paper, we propose to incorporate the gaze fixation prediction score into ColBERT when the query and the document interact. First, we use the Wordpiece tokenizer to tokenize the query and document separately and obtain two term sequences: $q_1, q_2, \dots, q_{qlen}$ and $d_1, d_2, \dots, d_{dlen}$. GazBy-b appends [Q] token to the input Wordpiece query tokens, [CLS] + [Q] + $q_1 \dots q_{qlen}$ + [SEP], following [33], and prepends a document indicator token, [D] to the document input tokens [CLS] + [D] + $d_1 \dots d_{dlen}$ + [SEP]. We perform query augmentation by padding the query terms with BERT’s [mask] tokens up to a predefined length of M_q and do not augment the documents. Second, we feed the tokenized q and d terms into the gaze fixation prediction model to obtain the gaze prediction for each term, $g(q_1), g(q_2), \dots, g(q_{qlen})$, and $g(d_1), g(d_2), \dots, g(d_{dlen})$.

Table 2: Statistics of TREC DL 2019-2020 passage retrieval.

	Training	Development	2019 Test	2020 Test
# of queries	502,939	6,980	43	54
# of qrels	532,761	7,437	9,260	11,386
# of passages	8,841,823			

Third, simultaneously with the second step, we feed the tokenized query and document into two separate BERT encoders and fully connected layers to get the contextual representations for each term: $\vec{q}_1, \dots, \vec{q}_{qlen}$ and $\vec{d}_1, \dots, \vec{d}_{dlen}$. Fourth, we calculate the relevance score $S(q, d)$ by using a modified *MaxSim* operator:

$$S(q, d) = \sum_i^{qlen} g(q_i) \cdot \max_j \cos(\vec{q}_i, \vec{d}_j) \cdot g(d_j), \quad (4)$$

We optimize GazBy-b by minimizing the pairwise cross-entropy loss the same as in Eq. 3.

Note that the gaze fixation scores are added when the q-d interaction takes place. To test if it can be done at other places, we also have a variation, GazBy-b Last Layer, such that it multiplies the gaze prediction to the attention scores at the last encoder layer. We also test adding the gaze prediction scores at both the attention layers and the *MaxSim* function (this setting is called GazBy-b Combined). The results are reported in Section 4.

4 EXPERIMENTS

This section reports our experiment results and findings for passage re-ranking on the TREC 2019 and 2020 Deep Learning (DL) Tracks.

4.1 Task and Datasets

The document collections used in TREC DL 2019-2020 Tracks are based on training data from MS MARCO, a dataset created by Microsoft in 2016 and adapted to ad hoc retrieval tasks in 2018 [44]. The datasets were created with the aim of improving ad hoc retrieval with training data that has sparse labels, mimicking “real-world” retrieval where the number of relevant documents through user click logs is sparse. MS MARCO is a collection of 8.8 million web passages and 1 million Bing user searches. No documents in the dataset are marked as irrelevant, and each query is associated with one or more positive passages.

In the pooling and judging process, the National Institute of Standards and Technology (NIST) chose a subset of these sampled queries for judging, based on budget constraints and with the goal of finding a sufficiently comprehensive set of relevance judgments to make the test collection reusable [5, 6]. This led to a judged test set of 43 queries in 2019 and 54 queries in 2020. The qrels file contains a four-point scale judgments from irrelevant (0) to perfect relevant (3). For metrics that binarize the judgment scale, we map passage judgment levels 2 and 3 to relevant and map document judgment levels 0 and 1 to irrelevant. We use TREC’s evaluation scripts to compute the above metrics on our retrieved results.² Table 2 shows the statistics of the dataset used in our experiments.

Without loss of generality, our experiments focus on the passage re-ranking task. We evaluate all baselines and proposed models on

²https://github.com/usnistgov/trec_eval.

the TREC DL 2019 and 2020 passage re-ranking task, for which we re-rank 1000 passages that were provided by NIST for each query.

4.2 Metrics

To evaluate the retrieval effectiveness, we employ TREC DL’s official evaluation metrics. They include: Precision (P) [8] at rank position 10 $P = \frac{1}{Q} \sum_{i=1}^{|Q|} \frac{\text{relevant documents retrieved}}{10}$, averaged over all test queries; Normalized Discounted Cumulative Gain (nDCG) at rank 10 [28] $nDCG_{10} = \frac{DCG_{10}}{IDCG_{10}}$, which is made up of $DCG_{10} = \sum_{i=1}^{10} \frac{2^{rel_i-1}}{\log_2(i+1)}$, the discounted cumulative gain, and $IDCG_{10} = \sum_{i=1}^{|REL_{10}|} \frac{rel_i}{\log_2(i+1)}$, the ideal discounted cumulative gain, where rel_i is the relevance of the result at position i and REL_{10} represents the list of relevant documents ordered by their relevance up to position 10; Mean Average Precision [8] $MAP = \frac{1}{Q} \sum_{i=1}^{|Q|} AP(q_i)$, averaged over all test queries; and Reciprocal Rank (RR) [8] $RR = \frac{1}{Q} \sum_{i=1}^{|Q|} \frac{1}{rank_i}$, the inverse of the first occurrence of the first relevant document in a list of results, averaged over all test queries.

4.3 Experimental Setup

4.3.1 Baselines:

- **BM25 [56]**: a traditional probabilistic retrieval model that is one of the top performing IR methods prior to deep learning. We use the Anserini toolkit³ with all default settings to reproduce the BM25 experiments.
- **MonoBERT [46]**: a cross-encoder neural retrieval model based on BERT Large. Our implementation follows [46] and uses a query-document concatenated sequence [CLS] + $x_1 \dots x_{q_{len}}$ + [SEP] + $x_1 \dots x_{d_{len}}$ + [SEP] as input to BERT Large to obtain q-d term embeddings. The embedding of the [CLS] token is used for classification after being fed into a feed forward neural network. Following [45], we use 1024 dimensional vectors to represent query and document token embeddings as the hidden states of our transformer layers, with 512 total tokens in the query and document input sequence. Our transformer has 16 attention heads in each encoder layer, and uses a dropout rate of 0.1 on attention probabilities. It is the basis for many top performing ad hoc retrieval models [5, 6].
- **ColBERT [33]**: a bi-encoder neural retrieval model based on BERT that uses a late interaction mechanism to improve query-time efficiency, where the query and document embeddings are learned separately and only interact with each other by MaxSim function at the last stage of retrieval. It is a highly effective method with slightly worse performance in effectiveness than MonoBERT but runs much more efficient. Our implementation follows [33].
- **ColBERT + tf-idf**: a variant of ColBERT that uses tf-idf to weight the MaxSim scores for each query term: $S(q, d) = \sum_i^{q_{len}} idf(q_i) \cdot MaxSim(\vec{q}_i, d)$. It uses tf-idf, a popular term weighting scheme.

³<https://github.com/castorini/anserini>.

4.3.2 Cross-encoder GazBy variations:

Methods merging gaze during q-d interaction:

- **GazBy-c (or GazBy-c Last Layer)**: the proposed variation of GazBy for cross-encoders as described in Section 3.3. We use MonoBERT [46] with BERT Large as our base model. The implementation details of BERT Large are as outlined in [46].
- **GazBy-c All Layers**: a variation of GazBy-c. It is identical to GazBy-c Last Layer, except that each encoder layer (24 total) uses gaze predictions to modify scaled dot product attention across 16 attention heads, instead of only modifying the last encoder layer.

Methods merging gaze before q-d interaction:

- **GazBy-c First Layer**: another variation of GazBy-c. It incorporates gaze prediction into the cross-encoder before it creates interaction between the (q,d) pair in the first encoder layer. We expand each gaze fixation prediction score generated by our gaze model to a dimension of 1024, such that each gaze fixation score is repeated 1024 times in the expanded vector, creating a $(q_{len} + d_{len}) \times 1024$ gaze fixation prediction tensor. We then use element-wise multiplication to multiply the input Wordpiece embedding vectors by this tensor. Finally, we feed these embeddings into BERT Large and score the relevance through the [CLS] token.

4.3.3 Bi-encoder GazBy variations:

Methods merging gaze during q-d interaction:

- **GazBy-b (or GazBy-b MaxSim)**: the proposed variation of GazBy for bi-encoders as described in Section 3.4. Following the default settings of ColBERT, we pad the query and document to the max length of 32 and 180, respectively. We use BERT Large as the encoder ; and the implementation details are identical for the variations below.

Methods merging gaze before q-d interaction:

- **GazBy-b Last Layer**: a variation of GazBy-b that performs interaction between the (q,d) pair before their interaction in the MaxSim operator. Instead of using gaze predictions as term weights as in GazBy-b MaxSim, we element-wise multiply the gaze fixation predictions with the attention scores of the last layer of query and document encoders. This incorporation of gaze is similar to what is done in “GazBy-c Last Layer” (See Eq. 2). Then the un-modified MaxSim function is used to compute the relevance score using the resulting query and document embeddings. Other pre-processing and training details are the same with GazBy-b.
- **GazBy-b Combined**: We also test adding the gaze prediction scores both at the attention layers and at the MaxSim function. We combine GazBy-b MaxSim and GazBy-b Last Layer. That is, we element-wise multiply the gaze fixation predictions with the attention scores at the last layer of query and document encoders and then use modified MaxSim function to get the relevance score.

Table 3: Passage re-ranking results on TREC DL 2019 and 2020, separated by whether the model was Bi-Encoder or Cross-Encoder based. The best performing results are shown in bold for each section. Two arrows indicate that the change in performance from the baseline ColBERT (for GazBy-b variants) or MonoBERT (for GazBy-c variants) model is greater than 5%, while one arrow indicates the direction of the change was no greater than 5%.

Method	DL 2019				DL 2020			
	P@10	nDCG@10	MAP	RR	P@10	nDCG@10	MAP	RR
BM25	0.412	0.506	0.301	0.704	0.350	0.480	0.286	0.659
MonoBERT	0.614	0.703	0.433	0.881	0.558	0.701	0.466	0.796
GazBy-c First Layer	0.028 ↓↓	0.048 ↓↓	0.068 ↓↓	0.069 ↓↓	0.015 ↓↓	0.031 ↓↓	0.032 ↓↓	0.060 ↓↓
GazBy-c All Layers	0.033 ↓↓	0.050 ↓↓	0.061 ↓↓	0.080 ↓↓	0.024 ↓↓	0.028 ↓↓	0.030 ↓↓	0.051 ↓↓
GazBy-c Last Layer	0.621 1.1%↑	0.717 2.0%↑	0.438 1.2%↑	0.881	0.544 -2.5%↓	0.696 -0.7%↓	0.444 -4.7%↓	0.832 4.5%↑
ColBERT	0.619	0.713	0.447	0.861	0.533	0.698	0.460	0.856
ColBERT tf-idf	0.614 -0.8%↓	0.707 -0.8%↓	0.443 -0.9%↓	0.831 -3.5%↓	0.535 0.4%↑	0.699 0.1%↑	0.461 0.2%↑	0.854 -0.2%↓
GazBy-b Last Layer	0.505 ↓↓	0.579 ↓↓	0.332 ↓↓	0.759 ↓↓	0.374 ↓↓	0.484 ↓↓	0.279 ↓↓	0.587 ↓↓
GazBy-b Combined	0.302 ↓↓	0.487 ↓↓	0.255 ↓↓	0.559 ↓↓	0.321 ↓↓	0.391 ↓↓	0.239 ↓↓	0.502 ↓↓
GazBy-b MaxSim	0.616 -0.5%↓	0.698 -2.1%↓	0.429 -4.0%↓	0.878 2.0%↑	0.541 1.5%↑	0.704 0.9%↑	0.460	0.846 -1.2%↓

4.3.4 Training, Validation, and Testing. In our experiments, the training data takes the form of triples: (query, positive passage, negative passage). These triples are generated from TREC’s provided training qrels. Validation triples are available in an identical format. We train all GazBy-c models and the MonoBERT baseline on 4k training triples for 4 epochs. Our model is based on the *Hugging Face* implementation of BERT for sequence classification.⁴ The MonoBERT checkpoint is loaded from castorini⁵ prior to fine-tuning, which has been trained on the MS MARCO document collection. For each epoch, we perform validation with a subset of 700 development triples to increase validation speed, and at prediction time we select the model with the highest validation accuracy. We train all GazBy-b models and the ColBERT baseline on the TREC DL train triples collection for 50,000 steps with batch size equals 32. All GazBy-c and GazBy-b models are trained using the Adam Optimizer with learning rate $3e^{-6}$ and $\epsilon = 1e^{-6}$. Three GeForce RTX NVIDIA GPUs are used for training.

4.4 Main results

Table 3 reports the main experimental results on search effectiveness for the official TREC DL 2019 and 2020 passage re-ranking tasks. From Table 3, we can see that “GazBy-c Last Layer” and “GazBy-b Maxsim” give the best performance among cross-encoder GazBy and bi-encoder GazBy variations, respectively. They also work well compared to their own baselines, MonoBERT and ColBERT. In particular, “GazBy-c Last Layer” outperforms MonoBERT on P@10, MAP, and nDCG@10 by 1-2%, while maintaining effectiveness in terms of RR on the TREC DL 2019 Dataset. On the TREC DL 2020 Dataset, “GazBy-c Last Layer” outperforms MonoBERT by 4.5% on RR, but decreases in terms of effectiveness by 2.5% on P@10 and 4.7% on MAP, and 0.7% on nDCG@10. Comparing with ColBERT on DL 2019, our “GazBy-b MaxSim” run improves performance by 2.0% on RR, but decreases performance by 2.0% on nDCG@10, 4.1% on MAP, and 1% on P@10. For DL 2020, it improves on the ColBERT baseline by 2% on P@10 and 1.2% on nDCG@10, and decreases 1.2% on RR. Given that both MonoBERT and ColBERT are highly effective retrieval models that boosted the SOTA

performance by significant margin from previous approaches, the improvements that we observe from Gazby are quite encouraging.

Note that MonoBERT and ColBERT’s high performance gain rely on large scale pre-trained model that is extensively trained using superior computational powers. In contrast, the gaze model that we add on top of them is rather light-weight. Our gaze prediction model is only trained over 6,000 sentences, and our gaze prediction model has far fewer parameters (302,593) than BERT Large (345 million). Given the limited resources that our proposed method requires and limited gaze training data available, the amount of improvements that GazBy has achieved show a promising new direction in neural information retrieval.

4.5 Our Findings

Through Table 3, we realize that some settings of GazBy works poorly. For instance, we find that incorporating gaze into every attention layer, “GazBy-c All Layers” by performing the modified scaled dot product attention in Eq. 2 decreases model performance (as compared to “GazBy-c Last Layer”) significantly by 96%. This decrease in performance shows that one cannot use gaze fixation scores in multiple places throughout the cross-encoder. We also use gaze fixation scores to modify the Wordpiece embeddings before the q-d interaction in “GazBy-c First Layer”. This method severely decreases the performance of “GazBy-c Last Layer” by up to 97%.

For the bi-encoder GazBy, we find that except incorporating gaze fixation scores during the interaction between query and document using the modified *MaxSim* operator, other settings perform poorly. For instance, incorporating gaze before the interaction, as in the case of “GazBy-b Last Layer” for DL 2019 in Table 3 decreases performance by up to 25% on P@10 and 20% on RR. Similar degrades are observed on DL 2020. “GazBy-b Combined,” which uses both Last Layer and MaxSim settings performs even worse.

These results show that even though gaze prediction can be a useful component to improve transformer-based retrieval models, they are sensitive to where the merging of the two components should happen. Based on what we observe from our experiments, the only effective merging point of the gaze model and the transformer model is when the query-document interaction happens.

⁴https://huggingface.co/docs/transformers/model_doc/bert.

⁵<https://huggingface.co/castorini/monobert-large-msmarco>.

5 CONCLUSION

In this paper, we investigate effective ways to incorporate human gaze attention into existing transformer-based neural information retrieval models that largely benefited from the recent development in the use of computational attention. Based on our experiments over the TREC DL 2019 and 2020 Tracks, a *key finding* is that the only effective merging point of human gaze attention and computational attention in the transformer models is when the query-document interaction happens in the retrieval algorithms.

We are intrigued by the experimental results that we observe. Based on our experiments, it is encouraging and effective to use gaze fixations to help with transformer-based retrieval models. In addition, how and where to combine the two have a significant impact on their joint effectiveness: It is not as simple as to say that we can add gaze fixation scores into the attention layers, which works for MonoBERT but not for ColBERT.

We did extensive experiments on models currently dictating the direction of the field. Two transformer-based retrievers, one interaction-based and one representation-based, are investigated. Our next immediate effort would be investigating our findings over other traditional and non-transformer neural retrievers, such as the recent dense retrievers and sparse retrievers.

We acknowledge that we make an assumption about our gaze data that may be incorrect. We assume our gaze data is well suited to the type of gaze humans apply to query and document text when determining whether they are relevant, as is the task in relevance scoring. This assumption was made out of necessity. Information retrieval is related to information need based attention. Such human attention are human attention patterns on a candidate text when the human subject is trying to understand whether that text is relevant to their needs. For example, a subject might use need-based attention when reading an article to determine whether it is relevant to a question about a related topic. Note that the human attention data we use to pre-train our model, in contrast, is comprehension based: subjects in the GeCo and ZuCo data sets are reading to understand their texts. Therefore, there is a difference between the type of the human attention data we use to pre-train our gaze prediction model and the type of attention an search engine users exhibit to determine query-document relevance. In light of this fact, our work seeks to be a starting point for future efforts. With the advent of AR/VR technology [60, 77], we are confident that devices and datasets that show the gaze of human subjects performing relevance scoring will be more available. In future work, we seek to improve GazBy by using such data (for instance, use gaze data for post-retrieval relevance feedback to neural retrievers), and to incorporate gaze prediction in non-transformer based models to understand where gaze can be best applied. The findings from this paper suggest that the interaction between query and document in these models would be a great starting place for such efforts.

ACKNOWLEDGEMENT

The authors are thankful for the ACM SIGIR/ICTIR student travel grant, the Royden B. Davis Fellowship, a Provost's Undergraduate Research Presentation Award, and the U.S. National Science Foundation MAPWISELY Mid-Career Fund.

REFERENCES

- [1] Samira Abnar and Willem Zuidema. 2020. Quantifying Attention Flow in Transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL '20)*. Online, 4190–4197. <https://doi.org/10.18653/v1/2020.acl-main.385>
- [2] Yang Bai, Xiaoguang Li, Gang Wang, Chaoliang Zhang, Lifeng Shang, Jun Xu, Zhaowei Wang, Fangshan Wang, and Qun Liu. 2020. SparTerm: Learning term-based sparse representation for fast text retrieval. *arXiv preprint arXiv:2010.00768* (2020).
- [3] Panos Balatsoukas and Ian Ruthven. 2012. An eye-tracking approach to the analysis of relevance judgments on the Web: The case of Google search engine. *Journal of the American Society for Information Science and Technology (JASIST)* 63, 9 (2012), 1728–1746.
- [4] Uschi Cop, Nicolas Dirix, Denis Drieghe, and Wouter Duyck. 2017. Presenting GECO: An eyetracking corpus of monolingual and bilingual sentence reading. *Behavior research methods* 49, 2 (2017), 602–615.
- [5] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2020. Overview of the TREC 2020 deep learning track. In *Proceedings of the 29th Text REtrieval Conference*. Gaithersburg, MA, USA. arXiv:2102.07662 [cs.IR]
- [6] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2019. Overview of the TREC 2019 deep learning track. In *Proceedings of the 28th Text REtrieval Conference (TREC '19)*. Gaithersburg, MA, USA. arXiv:2003.07820 [cs.IR]
- [7] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of Recommender Algorithms on Top-n Recommendation Tasks. In *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys '10)*. Barcelona, Spain, 39–46. <https://doi.org/10.1145/1864708.1864721>
- [8] W Bruce Croft, Donald Metzler, and Trevor Strohman. 2010. *Search engines: Information retrieval in practice*. Vol. 520. Addison-Wesley Reading.
- [9] Zhuyun Dai and Jamie Callan. 2020. Context-aware sentence/passage term importance estimation for first stage retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. 1533–1536.
- [10] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *Proceedings of the eleventh ACM international conference on web search and data mining (WSDM '18)*. 126–134.
- [11] 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) (NAACL '19)*. Minneapolis, Minnesota, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- [12] Sibó Dong, Justin Goldstein, and Grace Hui Yang. 2022. SEINE: SEGment-based Indexing for NEural information retrieval. In *Workshop of Reaching Efficiency in Neural Information Retrieval (ReNeuIR) of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ReNeuIR '22)*. Madrid, Spain.
- [13] Ralf Engbert, Antje Nuthmann, Eike M Richter, and Reinhold Kliegl. 2005. SWIFT: a dynamical model of saccade generation during reading. *Psychological review* 112, 4 (2005), 777.
- [14] Yixing Fan, Jiafeng Guo, Yanyan Lan, Jun Xu, Chengxiang Zhai, and Xueqi Cheng. 2018. Modeling diverse relevance patterns in ad-hoc retrieval. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*. 375–384.
- [15] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. 2288–2292. <https://doi.org/10.1145/3404835.3463098>
- [16] Luyu Gao and Jamie Callan. 2021. Condenser: a Pre-training Architecture for Dense Retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP '21)*. Online and Punta Cana, Dominican Republic, 981–993.
- [17] Alex Graves and Jürgen Schmidhuber. 2005. Framework phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks* 18, 5-6 (2005), 602–610.
- [18] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM international conference on information and knowledge management (CIKM '16)*. 55–64.
- [19] Jacek Gwizdka and Yinglong Zhang. 2015. Differences in eye-tracking measures between visits and revisits to relevant and irrelevant web pages. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '15)*. 811–814.
- [20] 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 (EMNLP '16)*. Austin, Texas, 85–95. <https://doi.org/10.18653/v1/D16-1009>

- [21] John M Henderson, Taylor R Hayes, Gwendolyn Rehrig, and Fernanda Ferreira. 2018. Meaning guides attention during real-world scene description. *Scientific reports* 8, 1 (2018), 1–9.
- [22] Nora Hollenstein and Lisa Beinborn. 2021. Relative Importance in Sentence Processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers) (ACL/IJCNLP '21)*. Online, 141–150. <https://doi.org/10.18653/v1/2021.acl-short.19>
- [23] Nora Hollenstein, Federico Pirovano, Ce Zhang, Lena Jäger, and Lisa Beinborn. 2021. 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 (NAACL '21)*. Online, 106–123. <https://doi.org/10.18653/v1/2021.naacl-main.10>
- [24] Nora Hollenstein, Jonathan Rotsztein, Marius Troendle, Andreas Pedroni, Ce Zhang, and Nicolas Langer. 2018. ZuCo, a simultaneous EEG and eye-tracking resource for natural sentence reading. *Scientific data* 5, 1 (2018), 1–13.
- [25] Nora Hollenstein, Marius Troendle, Nicolas Langer, Daniel Wiechmann, Dhruv Mullick, dayekim, Martyna B Plomecka, janetyan, and Yixuan Zhang. 2022. ZuCo 2.0: A Dataset of Physiological Recordings During Natural Reading and Annotation. osf.io/2urht
- [26] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management (CIKM '13)*. 2333–2338.
- [27] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. In *Proceedings of the 8th International Conference on Learning Representations (ICLR '20)*. <https://openreview.net/forum?id=SkxgnnNFvH>
- [28] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.
- [29] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.
- [30] Nour Kaessli, Zeynep Akata, Bernt Schiele, and Andreas Bulling. 2017. Gaze embeddings for zero-shot image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '17)*. 6412–6421. <https://doi.org/10.1109/CVPR.2017.679>
- [31] Vladimir Karpukhin, Barlas Ögüz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP '20)*. Online.
- [32] Frank Keller. 2010. Cognitively plausible models of human language processing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10)*. Uppsala, Sweden, 60–67.
- [33] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval (SIGIR '20)*. Xi'an, China, 39–48.
- [34] Sigrid Klerke and Barbara Plank. 2019. At a glance: The impact of gaze aggregation views on syntactic tagging. In *Proceedings of the Beyond Vision and Language: Integrating Real-world Knowledge (LANTErn '19)*. 51–61.
- [35] Dmitry Lagun and Eugene Agichtein. 2015. Inferring Searcher Attention by Jointly Modeling User Interactions and Content Salience. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '15)*. Santiago, Chile, 483–492. <https://doi.org/10.1145/2766462.2767745>
- [36] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL '19)*. Florence, Italy, 6086–6096. <https://doi.org/10.18653/v1/P19-1612>
- [37] Xiangsheng Li, Yiqun Liu, Jiaxin Mao, Zexue He, Min Zhang, and Shaoping Ma. 2018. Understanding Reading Attention Distribution during Relevance Judgement. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*. Torino, Italy, 733–742. <https://doi.org/10.1145/3269206.3271764>
- [38] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021. Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies* 14, 4 (2021), 1–325.
- [39] Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonello, Nazli Goharian, and Ophir Frieder. 2020. Expansion via Prediction of Importance with Contextualization. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. 1573–1576. <https://doi.org/10.1145/3397271.3401262>
- [40] Antonio Mallia, Omar Khattab, Torsten Suel, and Nicola Tonello. 2021. Learning passage impacts for inverted indexes. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. 1723–1727.
- [41] Jonathan Malmaud, Roger Levy, and Yevgeni Berzak. 2020. Bridging information-seeking human gaze and machine reading comprehension. In *Proceedings of the 24th Conference on Computational Natural Language Learning (CoNLL '21)*. Online, 142–152. <https://doi.org/10.18653/v1/2020.conll-1.11>
- [42] 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 (EMNLP '13)*. 803–807.
- [43] James Michaelov and Benjamin Bergen. 2020. How well does surprisal explain N400 amplitude under different experimental conditions?. In *Proceedings of the 24th Conference on Computational Natural Language Learning (CoNLL '20)*. Online, 652–663. <https://doi.org/10.18653/v1/2020.conll-1.53>
- [44] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@ NIPS 2016*.
- [45] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).
- [46] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424* (2019).
- [47] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, and Xueqi Cheng. 2016. A study of matchpyramid models on ad-hoc retrieval. *arXiv preprint arXiv:1606.04648* (2016).
- [48] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16)*. Phoenix, Arizona, 2793–2799.
- [49] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, and Xueqi Cheng. 2017. DeepPrank: A new deep architecture for relevance ranking in information retrieval. In *Proceedings of the 2017 ACM Conference on Information and Knowledge Management (CIKM '17)*. 257–266.
- [50] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP '14, Vol. 14)*. 1532–1543. <https://doi.org/10.3115/v1/D14-1162>
- [51] Tingting Qiao, Jianfeng Dong, and Duanqing Xu. 2018. Exploring human-like attention supervision in visual question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI '18)*. New Orleans, Louisiana, USA.
- [52] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL '21)*. Online, 5835–5847. <https://doi.org/10.18653/v1/2021.naacl-main.466>
- [53] Keith Rayner. 1977. Visual attention in reading: Eye movements reflect cognitive processes. *Memory & cognition* 5, 4 (1977), 443–448.
- [54] Erik D Reichle, Keith Rayner, and Alexander Pollatsek. 2003. The EZ Reader model of eye-movement control in reading: Comparisons to other models. *Behavioral and brain sciences* 26, 4 (2003), 445–476.
- [55] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. 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 '19)*. Hong Kong, China, 3982–3992. <https://doi.org/10.18653/v1/D19-1410>
- [56] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (apr 2009), 333–389. <https://doi.org/10.1561/15000000019>
- [57] Omid Rohanian, Shiva Taslimipour, Victoria Yaneva, and Le An Ha. 2017. Using gaze data to predict multiword expressions. In *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP '17)*. Varna, Bulgaria, 601–609. https://doi.org/10.26615/978-954-452-049-6_078
- [58] Jarkko Salojärvi, Ilpo Kojo, Jaana Simola, and Samuel Kaski. 2003. Can relevance be inferred from eye movements in information retrieval. In *Workshop on Self-Organizing Maps (WSOM '03, Vol. 3)*. 2003.
- [59] Iaroslav Shcherbaty, Andreas Bulling, and Mario Fritz. 2015. GazeDPM: Early Integration of Gaze Information in Deformable Part Models. *CoRR abs/15* (2015). <https://publications.cispa.saarland/1368/>
- [60] Mike Snider and Brett Molina. 2022. Everyone wants to own the metaverse including Facebook and Microsoft. But what exactly is it? *USA TODAY* (2022).
- [61] Ekta Sood, Simon Tannert, Philipp Müller, and Andreas Bulling. 2020. Improving natural language processing tasks with human gaze-guided neural attention. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS '20, Vol. 33)*. 6327–6341. <https://proceedings.neurips.cc/paper/2020/file/460191c72f67e90150a093b4585e7eb4-Paper.pdf>
- [62] Zhiwen Tang and Grace Hui Yang. 2019. DeepTileBars: Visualizing term distribution for neural information retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI '19, Vol. 33)*. 289–296.
- [63] Arun Balajee Vasudevan, Dengxin Dai, and Luc Van Gool. 2018. Object Referring in Videos With Language and Human Gaze. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '18)*.
- [64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all

- you need. In *Proceedings of the 31st Advances in neural information processing systems (NeurIPS '17)*. 5998–6008.
- [65] Aaron Veldre, Lili Yu, Sally Andrews, and Erik D Reichle. 2020. Towards a complete model of reading: Simulating lexical decision, word naming, and sentence reading with über-reader. In *Proceedings of the 42nd Annual Conference of the Cognitive Science Society*. Cognitive Science Society.
- [66] Zhiwei Wang, Yao Ma, Zitao Liu, and Jiliang Tang. 2019. R-transformer: Recurrent neural network enhanced transformer. *arXiv preprint arXiv:1907.05572* (2019).
- [67] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).
- [68] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end Neural Ad-hoc Ranking with Kernel Pooling. In *Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval (SIGIR '17)*. 55–64.
- [69] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *Proceedings of the 9th International Conference on Learning Representations, Virtual Event, Austria (ICLR '21)*. <https://openreview.net/forum?id=zeFrfgYZln>
- [70] Jia Xu, Lopamudra Mukherjee, Yin Li, Jamieson Warner, James M Rehg, and Vikas Singh. 2015. Gaze-enabled egocentric video summarization via constrained submodular maximization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2235–2244.
- [71] Victoria Yaneva, Le An Ha, Richard Evans, and Ruslan Mitkov. 2018. Classifying Referential and Non-referential It Using Gaze. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP '18)*. Brussels, Belgium, 4896–4901. <https://doi.org/10.18653/v1/D18-1528>
- [72] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '17)*. 490–498.
- [73] Kiwon Yun, Yifan Peng, Dimitris Samaras, Gregory J Zelinsky, and Tamara L Berg. 2013. Studying relationships between human gaze, description, and computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*. 739–746.
- [74] Hamed Zamani, Mostafa Dehghani, W Bruce Croft, Erik Learned-Miller, and Jaap Kamps. 2018. From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In *Proceedings of the 27th ACM international conference on information and knowledge management (CIKM '18)*. 497–506.
- [75] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. Repbert: Contextualized text embeddings for first-stage retrieval. *arXiv preprint arXiv:2006.15498* (2020).
- [76] Yingyi Zhang and Chengzhi Zhang. 2019. Using human attention to extract keyphrase from microblog post. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL '19)*. 5867–5872.
- [77] Andrew Jie Zhou and Grace Hui Yang. 2018. Minority Report by Lemur: Supporting Search Engine with Virtual Reality. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '18)*. Ann Arbor, MI, USA, 1329–1332. <https://doi.org/10.1145/3209978.3210179>
- [78] Shengyao Zhuang and Guido Zuccon. 2021. TILDE: Term Independent Likelihood MoDEL for Passage Re-Ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. 1483–1492. <https://doi.org/10.1145/3404835.3462922>