Perceived Text Relevance Estimation Using Scanpaths and GNNs

Abdulrahman Mohamed Selim
abdulrahman.mohamed@dfki.de
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany

Omair Shahzad Bhatti
omair_shahzad.bhatti@dfki.de
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany

Michael Barz
michael.barz@dfki.de
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany
University of Oldenburg
Oldenburg, Germany

Daniel Sonntag
daniel.sonntag@dfki.de
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany
University of Oldenburg
Oldenburg, Germany

ABSTRACT

A scanpath is an important concept in eye tracking that represents a person's eye movements in a graph-like structure. Passive gaze-based interfaces, in which users do not consciously interact using their eyes, typically interpret users' scanpaths to enable adaptive and personalised interaction. Despite the benefits of graph neural networks (GNNs) in graph processing, this technology has not been considered for that purpose. An example application is perceived relevance estimation, which still suffers from low classification performance. In this work, we investigate how and whether GNNs can be used to analyse scanpaths for readers' perceived relevance estimation using the gazeRE dataset. This dataset contains eye tracking data from 24 participants, who rated the relevance of 12 short and 12 long documents in relation to a given query. The relevance was assigned either to an entire short document or to each paragraph within a long document, which allowed us to investigate two different GNN tasks. For comparison, we reproduced the gazeRE baseline using Random Forest and Support Vector classifiers, and an additional Convolutional Neural Network (CNN) from the literature. All models were evaluated using leave-users-out cross-validation. For short documents, the GNNs surpassed the baseline methods, with certain experiments showing an absolute balanced accuracy improvement of 7.6% and 14.3% over the CNN and gazeRE baselines, respectively. However, similar improvements were not observed in long documents. This work investigates and discusses the future potential of using GNNs as a scanpath analysis method for passive gaze-based applications, such as implicit relevance estimation.

CCS CONCEPTS

• **Computing methodologies → Neural networks**; • **Human-centered computing → Human computer interaction (HCI)**.

KEYWORDS

Eye Tracking; Scanpath; GNN; Passive Gaze-based Application

ACM Reference Format:
Abdulrahman Mohamed Selim, Omair Shahzad Bhatti, Michael Barz, and Daniel Sonntag. 2024. Perceived Text Relevance Estimation Using Scanpaths and GNNs. In *INTERNATIONAL CONFERENCE ON MULTIMODAL INTERACTION (ICMI '24), November 04–08, 2024, San Jose, Costa Rica.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3678957.3685736

1 INTRODUCTION

Scanpaths refer to traces of a person's eye movement across space over a period of time [19]. A scanpath consists of a series of alternating fixations and saccades [4]. Fixations describe the state when the eyes remain relatively still for a time period lasting somewhere between a few tens of milliseconds up to a few seconds, while saccades are the rapid eye movements from one fixation to another [19]. Scanpaths are among the most common methods for analysing and representing human eye movements [4, 27]. Figure 1 is an example of a visual encoding format where a scanpath is projected on top of a stimulus, e.g. a piece of text, where fixations are shown as numbered circles, and saccades are lines connecting them. Graph representation is another common format, where a scanpath gaze data is grouped, e.g. clustering neighbouring fixations, to create nodes and edges representing a graph structure [30].

Graph Neural Networks (GNNs) are deep-learning models that process graph structures and capture their dependence via message passing between nodes [43]. GNNs have shown good performance in multiple fields, e.g., natural science, social science, and bioinformatics [43]. Despite this, GNNs have not been properly investigated in processing scanpath graph structures. There have been attempts in active gaze-based applications, e.g. [35]. However, we only found one publication, i.e. [38], that used GNNs with scanpath data in a passive gaze-based application, i.e., applications where eye tracking is used as a supporting modality to understand a user's behaviour and activities without explicit gaze-based interaction [13, 34].

An important area for passive gaze-based applications is to understand and monitor cognitive processes [29], e.g. implicit relevance estimation during reading [1, 2, 6] or during decision making [15]. Detecting a user's perceived relevance towards a piece of media is often used to improve the system performance and return user-tailored results, e.g., in recommender [33] and information

ICMI '24, November 04–08, 2024, San Jose, Costa Rica

ACM ISBN 979-8-4007-0462-8/24/11
https://doi.org/10.1145/3678957.3685736

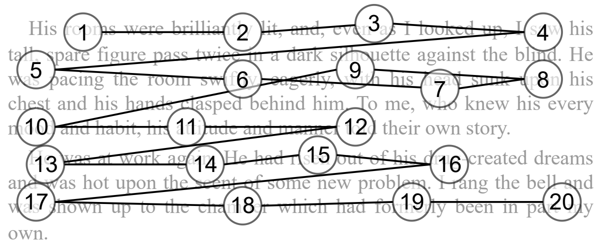


Figure 1: A manually generated scanpath over a piece of text as a simplified version of a real-world example.

retrieval [32] systems. Perceived relevance estimation is also used in human-computer interaction (HCI), e.g. to create adaptive user interfaces (UIs) [14, 15]. However, explicitly detecting perceived relevance using questionnaires or interviews could have a negative impact on a user's cognitive load [33], which is why implicit relevance estimation is seen as a better alternative because it requires no extra effort on a user's behalf [33, 39].

In this paper, we investigated using GNNs for scanpath processing to estimate a user's perceived relevance while reading text documents using the gazeRE dataset [1]. This dataset contains data from two different tasks using two different text corpora, where a user's perceived relevance to a given query is estimated either for each paragraph in a document or for the document as a whole. This enabled us to treat each full document as a single graph and investigate a graph classification task, i.e. predicting the label assigned to the full document, and a node classification task, i.e. predicting the label assigned to each single paragraph. Using a GNN requires converting scanpaths into suitable graph structures. We implemented and compared four different scanpath graph structures to use as inputs to our GNNs. To evaluate our proposed method, we implemented two baseline approaches. The first approach reproduced the setup of Barz et al. [1], which used 17 eye tracking features with random forest (RF) and support vector machine (SVM) classifiers. We extended this setup by investigating an additional feature subset. The second approach replicated the method of Bhattacharya et al. [3], which used the VGG19 Convolutional Neural Network (CNN) architecture [36]. We extended this approach by examining its performance not only on short documents, similar to those used in their original experiment, but also on long documents.

We designed our experiments to explore two primary research topics in a single, coherent framework. The first topic examined the feasibility of using GNNs to process scanpaths for perceived relevance estimation, focusing on both graph and node classification tasks. The second topic involved a comparative analysis of the performance of GNNs with other machine learning algorithms, namely SVM, RF, and CNN, in the context of perceived relevance estimation. This analysis aimed to provide a comprehensive understanding of the performance of these different approaches.

2 RELATED WORK

In the literature, we found different methods to construct graph structures out of scanpaths for passive gaze-based applications. Lan et al. [25] used a CNN to process complex graph structures where each gaze point represented a node for a stimulus and task inference application. Ma et al. [28] treated each word in a reading task as a node to structure a scanpath as a graph to measure reading comprehension using network metrics such as density, centrality, and small-worldness. Cantoni et al. [8] focused on modelling user viewing behaviour for user authentication by splitting the stimuli into 7x6 grids and used the centre of each grid cell to combine the different fixations into graph nodes; each node had a weight representing the total number of fixations and total fixation duration within it. Khosravan et al. [21] used the BIRCH clustering algorithm [42] to generate a less dense graph structure out of scanpaths on medical images to simplify the scanpaths without changing their topology; they encoded the number of nodes in each cluster and the total duration spent within each cluster in the graph as a representation for the attention in a particular region. Despite structuring scanpaths as graphs being common for passive gaze-based applications, we only found one paper by Wang et al. [38] that proposed a gaze-guided GNN to process graphs created by embedding the raw gaze data with image patches from x-ray scans.

GNNs have emerged as a powerful tool for learning with graph-structured data, such as molecules and social, biological, and financial networks; the key to this learning process is the effective representation of the graph structure [41, 43]. GNNs operate on a message passing scheme; each node calculates a new feature vector that contains structural information of its neighbouring nodes by aggregating the feature vectors of these neighbouring nodes; to represent an entire graph, a pooling method is used, such as summing the representation vectors of all nodes in the graph [41]. Zhou et al. [43] described a general GNN task pipeline, which consists of: defining the graph structure; determining the graph type (e.g., directed or undirected graph); identifying the task type, whether node-level tasks that focus on the graph nodes (e.g., node classification), edge-level tasks that focus on the graph edges (e.g., predicting if an edge exists between two nodes), or graph-level tasks that focus on the full graph structure (e.g., graph classification); and finally, building the GNN model. We wanted to combine GNNs and scanpath graph representations (without stimulus information) for a passive gaze-based application. The application we decided to focus on was estimating a user's perceived relevance while reading to see whether this approach could help improve the field's current state.

Previous studies showed that eye tracking is a valid modality for estimating a person's perceived relevance towards a text document with respect to a previously read trigger question. Buscher et al. [6] investigated the relation between a user's reading behaviour and their perceived relevance towards a document. They found that users tend to skim irrelevant documents but exert continuous reading behaviour while reading relevant ones. Gwizdka [17, 18] introduced the g-REL corpus, which is a collection of short text paragraphs and corresponding questions. They used it to investigate the relation between eye movements and a user's perceived relevance while reading and were able to confirm the prior findings of Buscher et al. [6].

Bhattacharya et al. [3] used the g-REL corpus and encoded users' scanpath data as images to estimate their perceived relevance using a CNN. They evaluated six different pre-trained CNNs but concluded that VGG19 [36] produced the best results. Afterwards, Bhattacharya et al. [2] introduced two novel convex hull-based scanpath features to estimate a user's perceived relevance while reading short news articles. They conducted two separate data collection studies where 24 participants and 120 news articles were used in the first study, and 24 participants and 42 news articles were used in the second study. They used 10-fold cross-validation with an RF classifier for a binary classification over three separate subsets, i.e. *Agree* where the user's perceived relevance matched the system relevance, *Topical* where the news articles were on the topic of interest but did not have the required answer, and *All* where they used the dataset as a whole. They achieved the best classification performance when they combined their two proposed convex hull features with 15 other eye tracking features from the literature. They reported the best model performance for the *Agree* subset followed by *All*, but the *Topical* subset produced poor results.

Barz et al. [1] extended the prior work of Bhattacharya et al. [2] by investigating using the same 17 features on long documents. They collected data from 24 participants using 12 documents from the g-REL corpus and 12 documents from the Google Natural Questions (GoogleNQ) corpus [24], which is a collection of long documents that require scrolling. Despite the lower model performance, using RF and SVM classifiers, they produced similar findings to Bhattacharya et al. [2] under the same experiment conditions for the g-REL corpus, but were unable to generalise their findings to the GoogleNQ corpus. Perceived relevance estimation is still an ongoing area of passive gaze-based research. It still has open questions regarding topical and long documents, so it is a suitable application domain to investigate and test GNNs for scanpath processing.

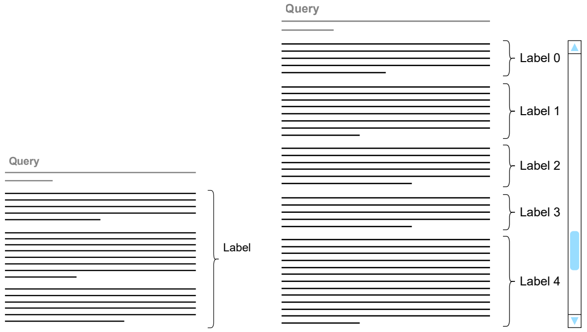
3 METHODS

In this paper, we present a novel GNN-based scanpath analysis approach using the gazeRE dataset for a node and a graph classification problem. For the graph classification, we evaluated four different scanpaths graph representation formats. However, for the node classification, we evaluated one graph representation format. We evaluated different GNN operators for both tasks. As a baseline, we reproduced the setup reported in [1] using SVM and RF classifiers; we also evaluated these classifiers using only the two convex hull-based features from [2]. In addition, to compare against a neural network, we replicated the VGG19 setup reported in [3].

3.1 Dataset

The gazeRE dataset[1] has eye tracking data from 24 participants for perceived relevance estimation while reading. Each participant read 12 short articles from the g-REL corpus [17] and 12 long articles from the GoogleNQ corpus [24].

The **g-REL** corpus contains four relevant, four irrelevant, and four topical documents with respect to their accompanying query according to the system label. Each document had between three to five paragraphs (μ = 3.5, σ = 0.645). Participants were shown a query and had to decide whether the entire document was *relevant* or *irrelevant* with respect to the query. A schematic example of a g-REL stimulus is shown in Figure 2a. Out of 288 total trials, 107 were perceived as relevant and 181 as irrelevant by the participants. The *Agree* subset (where the perceived relevance matched the system relevance) has 181 total trials, with 86 relevant and 95 irrelevant trials. The *Topical* subset (i.e., documents on the topic of interest but not containing the query answer) has 96 total trials, with 20 relevant and 76 irrelevant trials.



(a) g-REL Corpus (b) GoogleNQ Corpus

Figure 2: Schematic example of a stimulus for both g-REL in 2a and GoogleNQ in 2b from the gazeRE dataset [1].

The **GoogleNQ** corpus contains 12 long documents that require scrolling. One paragraph in each document is relevant, and the remaining paragraphs are topical to the accompanying query. GoogleNQ does not have explicitly irrelevant paragraphs. Each document had between five to seven paragraphs (μ = 5.83, σ = 0.799). Participants were shown a query and had to decide whether each separate paragraph in a document was *relevant* or *irrelevant* with respect to the query. A schematic example of a GoogleNQ stimulus is shown in Figure 2b. GoogleNQ has a total of 288 trials, with 450 relevant and 1230 irrelevant paragraphs. The *Agree* subset (where the full document perceived relevance matched the system relevance) has 145 total trials, with 248 relevant and 1190 irrelevant paragraphs. Due to all 12 documents having topical paragraphs, GoogleNQ did not have a *Topical* subset.

We used the *participants' perceived relevance* of the text to the query, i.e. relevant or irrelevant, as our labels for the binary classification problem. When we mention the word label moving forward, that is what we are referring to. There are multiple differences between g-REL and GoogleNQ. Each document in g-REL has one label assigned to the full document. However, in GoogleNQ, each document has multiple labels corresponding to the number of paragraphs within each document. Additionally, GoogleNQ does not have explicitly irrelevant paragraphs because all the paragraphs that do not have the answer to the query are on the topic of the query, i.e. Topical. We decided to use this dataset because the differences between the two corpora allowed us to investigate two different GNN task types: Node-level and Graph-level tasks. In both tasks, we treated each full document as a single graph. g-REL was suitable for graph classification because each document had one label assigned to it. GoogleNQ, on the other hand, was suitable for node classification, where we tried to classify the labels assigned to each paragraph in a document.

3.2 Traditional Machine Learning

In order to establish a baseline for comparison, we reproduced the setup reported in [1]. However, in addition to using the same 17

[1] https://github.com/DFKI-Interactive-Machine-Learning/gazeRE-dataset

Table 1: Overview of the 17 eye tracking features from [1, 2].

Type	Features
Fixation	**1.** Number of fixations **2.** Sum of fixation durations **3.** Mean of fixation durations **4.** Standard deviation of fixation durations
Saccade	**5.** Sum of horizontal amplitudes of saccades, normalised by w **6.** Sum of vertical amplitudes of saccades, normalised by h **7.** Sum of Euclidean distance of normalised saccade amplitudes **8.** Ratio of horizontal to vertical amplitudes **9.** Average saccade amplitude **10.** Horizontal saccade velocity **11.** Vertical saccade velocity **12.** Saccade velocity
Area	**13.** Area scanned by summed saccade amplitudes **14.** The scanned area normalised by the scan time **15.** Number of fixations per scanned area **16.** The convex hull area normalised by the scan time **17.** Number of fixations per convex hull area

features, shown in Table 1, we investigated using just the convex hull features from Bhattacharya et al. [2], i.e. numbers 16 and 17 in Table 1 because Bhattacharya et al. [2] only evaluated them on short documents and not longer documents such as the GoogleNQ corpus. We used three machine learning algorithms with our two feature sets: the default **RF** classifier from scikit-learn[2]; the **RF*** classifier which is the RF classifier with two additional preprocessing steps, the oversampling technique SMOTE [9] from the imbalanced-learn package [26], and the standardisation feature scaling method to make the features have zero mean and unit variance; in addition to the **SVM*** classifier which is the default SVM classifier from scikit-learn with the same preprocessing steps of RF*.

3.3 Convolutional Neural Network

We replicated the best-performing setup reported by Bhattacharya et al. [3] using VGG19, which is a variant of the VGG model with 19 layers. It includes 19 convolutional layers to capture the spatial patterns in images. The architecture uses small 3x3 convolution filters, which allow it to collect more detailed and complex features. In addition, it incorporates three fully connected layers following these convolutional layers. We adapted the final output layers to ensure their compatibility with our binary classification task.

In the preprocessing step, we transformed each eye tracking recording into a scanpath image following the methods used by Bhattacharya et al. [3]. For g-REL, each document produced one single image, while for GoogleNQ, we produced a scanpath image for each paragraph independently to ensure that the image dimensions and presentation remained consistent with those used for g-REL. An example of a scanpath image is shown in Figure 3, where each fixation is represented by unique visual markers proportional to its duration, while the saccades are colour-coded to illustrate the sequence of reading movements across the text. Each scanpath image was generated on a 2560x1440 canvas and scaled down to 256x256 as input to the CNN.

[2] https://scikit-learn.org/stable/

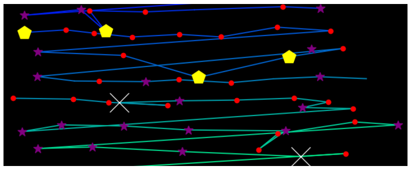


Figure 3: An example of a scanpath visual representation for the CNN. The fixations are represented by different markers based on their duration, while the saccades are colour-coded based on their timestamp.

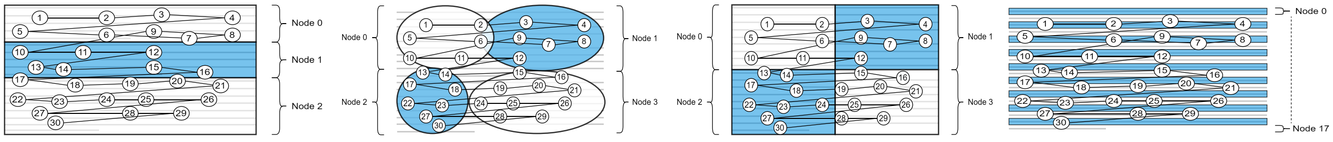
3.4 Graph Neural Network

3.4.1 Scanpath Graph Representation. In order to use the scanpaths as inputs to our GNNs, we converted the scanpaths into simplified graph structures. The generated graphs were directed (to retain the temporal information of a scanpath) and homogeneous (i.e., all the nodes and all the edges had the same type). The GoogleNQ corpus had a simpler conversion process because its documents were used in the node classification task; we treated each document as a single graph, each paragraph represented a node, and the saccades from one paragraph to the next represented the edges. However, each document in the g-REL corpus was accompanied by only one label; we tested four different approaches to generate suitable graphs from the scanpaths: *paragraph-based*, *line-based*, *cluster-based*, and *quartile-based*.

The *Paragraph-based* approach, shown in Figure 4a, is the same approach followed in GoogleNQ (to have a common representation between both corpora) where each paragraph represented a node, and the saccades from one paragraph to the next represented the edges. The *Line-based* approach, shown in Figure 4d, tries to preserve the structure of the text and reading patterns, which could be seen as an extension to Ma et al. [28] where they treated each word as a node. The *Cluster-based* approach was inspired by Khosravan et al. [21], but instead of using the BIRCH clustering algorithm [42], we used the Affinity Propagation algorithm [16]; Figure 4b shows a very simplified depiction of this approach. The *Quartile-based* approach divides the full-text document into four equal-sized nodes, as shown in Figure 4c; this is similar to splitting the stimuli into grids as reported by Cantoni et al. [8].

The documents in both corpora contained additional white space around the text along the X-axis. We ignored this extra white space, focusing only on the main body of the text. We limited the gaze points to the text body and not the background or document title.

Across the different graph generation approaches, we used the number of fixations and total fixation duration within each node as node features similar to [8, 21]. In addition, we computed the same 17 features shown in Table 1 for each node for all graph generation methods except for the line-based approach because we could not compute the area-based features and only computed the fixation and saccade-based features. To be able to use graph structures as inputs to a GNN, we need to define two parameters: Node Features (x) and Edges (E). The parameter x, shown in Algorithm 1, contains the node features, where each node has a feature vector. The parameter E, shown in Algorithm 1, contains the edges in the graph, which are directed in our use case.



(a) Paragraph-based **(b) Cluster-based** **(c) Quartile-based** **(d) Line-based**

Figure 4: Schematic examples of our four scanpath graph representations where each coloured element is a different node.

Algorithm 1: Graph Definition

```
Let $Graph = (V, E)$ be a directed graph, where $V$ is the set of nodes and $E$ is the set of edges.
$V = \{node^{(0)}, node^{(1)}, \ldots, node^{(n-1)}\}$ for $n$ nodes.
$E = \{(node^{(i)}, node^{(j)})\}$ for each edge from $node^{(i)}$ to $node^{(j)}$.
$x = (x^{(0)}, \ldots, x^{(n-1)}) = ((f_0^{(0)}, \ldots, f_m^{(0)}), \ldots, (f_0^{(n-1)}, \ldots, f_m^{(n-1)}))$ for $n$ nodes and $m$ node features.
Each $node^{(i)}$ is represented by $x^{(i)}$.
All connections from $node^{(i)}$ to other nodes in the graph are represented by edges $(node^{(i)}, node^{(j)})$.
For directed graphs $(node^{(i)}, node^{(j)}) \neq (node^{(j)}, node^{(i)})$.
```

3.4.2 GNN Model Architectures. We used the same GNN network architectures with both the graph and node classification problems. Wu et al. [40] stated in their review that an open research question is whether using deeper GNNs is actually a good strategy for learning graph data because the performance of some networks tended to drop with an increase in the number of graph convolutional layers. We kept our networks simple to investigate whether basic network structures could yield meaningful results and insights. We used PyTorch Geometric (PyG) for our implementations and used their documentation [3] as a starting point.

We used the Adam Optimiser [22] and the Cross-entropy Loss in our GNN architecture. Due to Cross-entropy Loss in PyTorch[4] already having a Sigmoid Activation function, we did not add an extra activation function. The graph classification GNN is shown in Figure 5a, while the node classification GNN is shown in Figure 5b. We had two main differences between the graph and node classification networks: (1) for graph classification, we used an additional readout layer, i.e. Global Average Pooling, which produces a single global representation for each graph from its nodes graph for the graph classification problem, and (2) we used different normalisation strategies between both problems. According to Chen et al. [10], graph classification problems perform better when the node features are normalised using batch-based normalisation, while node classification problems perform better when the features are normalised on a graph-based normalisation. We used BatchNorm [20] as our batch-based normalisation for the graph classification, and GraphNorm [7] as our graph-based normalisation for the node classification.

We evaluated various graph convolutional operators such as those from Morris et al. [31] and Kipf and Welling [23][5]. However, using the GATv2 operator from Brody et al. [5] performed the best, and we only focus on it here. The standard graph attentional (GAT) operator [37] assigns a weight, i.e. an attention coefficient, to each node's neighbours, which indicates the importance of each neighbouring node, and by using multiple attention heads, each head can learn a different type of information concerning the neighbourhood. While the GAT operator is computationally efficient, it has static attention, meaning the weights are fixed and cannot adapt based on the query or context. To address this, we utilised the GATv2 operator, which allows for dynamic changes in the weights. This flexibility enables the model to adapt better and has been shown to outperform the traditional GAT operator. Our networks, as shown in Figure 5, consisted of three GATv2Conv layers, each followed by an ELU activation function because GATv2 incorporates LeakyReLU in its computations, a Dropout function before the last layer to prevent overfitting, and a final Linear layer, which mapped the outputs from the convolution layers to the number of classes.

4 EXPERIMENT

In all our experiments, we split g-REL into *All*, *Agree*, and *Topical* subsets, and GoogleNQ into *All* and *Agree* subsets, similar to [1, 2]. *All* contained the whole dataset; *Agree* contained the data where the user's perceived relevance matched the system relevance; and *Topical* contained the data where the text was on the topic of interest, but without having the correct query answer.

We implemented a 5-fold stratified leave-users-out cross-validation using scikit-learn[6] to split the data into non-overlapping training and testing subsets. We used leave-users-out cross-validation because, for physiological data, traditional k-fold cross-validation might lead to overestimating the model performance [11, 12]. For each fold with the traditional machine learning models (i.e., RF and SVM), 80% of the data were used for training, and 20% were used for testing. However, with the CNN and GNN models, we used nested cross-validation for hyperparameter optimisation using the Optuna framework[7]. The outer cross-validation loop split the data into 20% for testing, and an inner cross-validation loop split the remaining 80% of the data into 64% for training and 16% for validation. In the inner cross-validation loop, each model configuration was tested on five different training and validation data splits, and then the validation performance metrics were averaged over the five folds. The

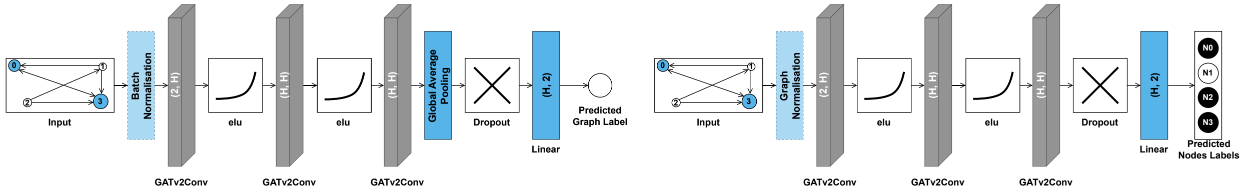
[3] https://pytorch-geometric.readthedocs.io/en/latest/get_started/colabs.html

[4] https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html

[5] Their respective architectures and results are available in the Appendix in the supplementary material.

[6] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedGroupKFold.html

[7] https://optuna.org/



(a) The Graph Classification Networks **(b) The Node Classification Networks**

Figure 5: Our Graph Convolution Neural Networks

configuration with the best average performance metric was then used to retrain the model on the whole training/validation data and produce the testing results; this was repeated five times for each testing subset. The pseudocode, shown in Algorithm 2, summarises this process. We used 5-fold leave-users-out cross-validation because five is a commonly used value, it is computationally efficient for nested cross-validation, and our data is quite small for two 10-fold cross-validation loops. Our codes are publicly available on GitHub[8].

We used balanced accuracy as our main evaluation metric. Balanced accuracy is the average of the sensitivity (i.e., the true positive rate or how many positive labels were correctly classified as positive) and the specificity (i.e., the true negative rate or how many negative labels were correctly classified as negative). Our goal is to correctly identify a user's perceived relevance towards a piece of text, which means focusing on correctly identified labels, whether *relevant* or *irrelevant*. Due to the data imbalance, especially for the *Topical* subset, we computed the balanced accuracy because it gives equal weight to both positive and negative classes. However, Barz et al. [1] used f1-score, which is the harmonic mean of the sensitivity and the precision (i.e., how many correct positive predictions exist in the total positive predictions). The issue is that the f1-score does not take into consideration the amount of true negative classifications, which is why we decided to use balanced accuracy as the main evaluation metric instead.

4.1 Results

Table 2 shows the test balanced accuracies averaged across the 5-fold stratified leave-users-out cross-validation. For each corpus subset, the result of the best-performing model is emphasised and underlined. In the Appendix[9], we reported additional performance metrics, such as f1-score, true positive rate, false positive rate, and area under the curve.

4.1.1 Traditional Machine Learning. For **g-REL**, using all 17 features with RF resulted in a 0.624 balanced accuracy for *All*. Both feature subsets resulted in an almost identical balanced accuracy of 0.692 for *Agree* using SVM*. On average, using all 17 features produced better results for *All* and *Agree*. None of the approaches produced balanced accuracies above 0.6 for *Topical*. For **GoogleNQ**, using all 17 features with SVM* resulted in a 0.604 balanced accuracy for *All*, but none of the approaches produced balanced accuracies above 0.6 for *Agree*.

Algorithm 2: GNN and CNN Model Training and Evaluation Using Nested Cross-validation

```
Input: Scanpath Data
Output: Average Test Data Balanced Accuracy
for each fold $i \in \{1, 2, 3, 4, 5\}$ do
    Split input data into training/validation set $D^{i}_{\text{train, val}}$ and test set $D^{i}_{\text{test}}$
    for Optuna trial $t \in \{1, 2, \ldots, n_{trials}\}$ do
        Pick model configuration $c^{i}_{t}$
        for each fold $j \in \{1, 2, 3, 4, 5\}$ do
            Split $D^{i}_{\text{train, val}}$ into training set $D^{ij}_{\text{train}}$ and validation set $D^{ij}_{\text{val}}$
            Train Model $m^{ij}_{ct}$ using configuration $c^{i}_{t}$ and training set $D^{ij}_{\text{train}}$
            Evaluate Model $m^{ij}_{ct}$ on validation set $D^{ij}_{\text{val}}$ to get the Balanced Accuracy $BA^{ij}_{\text{val},ct}$
            Store model configuration $c^{i}_{t}$, and Balanced Accuracy $BA^{ij}_{\text{val},ct}$
        Compute Average Validation Balanced Accuracy $\overline{BA}^{i}_{\text{val},ct}$ for configuration $c^{i}_{t}$
    Determine the configuration $c^{i}_{\text{best}}$ with the maximum $\overline{BA}^{i}_{\text{val},ct}$
    Train the best model $m^{i}_{\text{best}}$ using $c^{i}_{\text{best}}$ and $D^{i}_{\text{train, val}}$
    Test $m^{i}_{\text{best}}$ on $D^{i}_{\text{test}}$ to get test Balanced Accuracy $BA^{i}_{\text{test}}$
    Store $c^{i}_{\text{best}}$, and $BA^{i}_{\text{test}}$
Compute Average Test Balanced Accuracy $\overline{BA}_{\text{test}}$
```

4.1.2 Convolutional Neural Network. For **g-REL**, the CNN approach produced average balanced accuracies of 0.676 and 0.768 for *All* and *Agree*, respectively. For **GoogleNQ**, the CNN approach produced a 0.603 balanced accuracy for *Agree*. However, for the remaining subsets from both corpora, the CNN resulted in balanced accuracies below 0.6.

4.1.3 Graph Neural Network. For the graph classification using **g-REL**, the 17 node features resulted in higher balanced accuracies for both *All* and *Agree* compared to using only two node features.

[8] https://github.com/DFKI-Interactive-Machine-Learning/GNN-Scanpath-Analysis-ICMI2024

[9] In the supplementary material.

The paragraph-based scanpath graph representation produced the highest average balanced accuracies of 0.701 and 0.691 for *All* and *Agree*, respectively. However, for *Topical*, the cluster-based scanpath graph representation resulted in a 0.648 average balanced accuracy using only two node features, as opposed to 0.621 using all 17 node features.

For the node classification using **GoogleNQ**, the 17 node features were slightly better for both corpora. However, the average balanced accuracies were below 0.6 with 0.553 and 0.559 for *All* and *Agree*, respectively.

5 DISCUSSION

In this study, we implemented a GNN to process scanpath data for perceived relevance estimation, focusing on both graph and node classification tasks using the gazeRE dataset. We used established methods from the literature as baselines, comparing our GNN results with those obtained using traditional and neural network machine learning algorithms, namely SVM, RF, and CNN classifiers. The experiments were conducted with two primary objectives: (1) to assess the effectiveness of GNNs in scanpath analysis for perceived relevance estimation and (2) to compare the performance of GNNs with that of established methods from the literature.

5.1 Traditional Machine Learning

For **g-REL**, using all 17 features resulted in better accuracies than just the two convex hull features, which aligns with the findings from Bhattacharya et al. [2]. Our results were also consistent with the results from Barz et al. [1], who reported best balanced accuracies of 0.605 for *All*, 0.689 for *Agree*, and 0.527 for *Topical*. This difference in performance can be attributed to Barz et al. [1] using normal k-fold instead of leave-users-out cross-validation. Overall, *Agree* performed better than *All*, and none of the approaches produced meaningful results for *Topical*, which aligns with the findings from both Bhattacharya et al. [2] and Barz et al. [1].

For **GoogleNQ**, Barz et al. [1] were only able to achieve a 0.57 and a 0.543 balanced accuracy for both *All* and *Agree*, respectively. For *Agree*, we achieved similar results to Barz et al. [1]. However, for *All*, we were able to achieve a better balanced accuracy using all 17 features. The SVM* was the only approach, across all experiments, including the GNN and CNN approaches, to reach a balanced accuracy above 0.6 for *All*. The two convex hull features were unsuccessful in producing any meaningful results for either subset. Overall, despite the improvement for *All*, we believe it warrants further research as we cannot conclude the success of this approach at predicting users' perceived relevance with longer text documents.

5.2 Convolutional Neural Network

For **g-REL**, the CNN performed quite well. It resulted in 0.676, 0.768, and 0.572 balanced accuracies for *All*, *Agree*, and *Topical*, respectively. With a 5.2% and a 7.2% absolute difference for *All* and *Agree* compared to the traditional machine learning classifiers, which is a noticeable improvement. However, despite the improvement for *Topical*, its balanced accuracy is still below 0.6. Overall, we believe that the CNN was successfully replicated on a new dataset. This proves its ability to predict users' perceived relevance, but on short text documents.

For **GoogleNQ**, the CNN produced the overall best results for the *Agree* subset with a 0.603 balanced accuracy. However, it did not produce any meaningful results for *All*. Therefore, we cannot reach a proper conclusion regarding its success at predicting users' perceived relevance with longer text documents.

5.3 Graph Neural Network

For the GNN experiments, we start with the graph classification task using **g-REL**, and then we discuss the node classification task using **GoogleNQ**.

5.3.1 *Graph Classification Task.*

For **g-REL**, the GNN outperformed CNN and traditional machine learning for both *All* and *Topical*. It achieved a 0.701 balanced accuracy for *All* using the paragraph-based graph representation with 17 node features; this is a 2.5% and a 7.7% improvement over CNN and traditional machine learning, respectively. For *Topical*, it achieved a 0.648 balanced accuracy using the cluster-based graph representation with two node features, which is a 7.6% and a 14.3% improvement over CNN and traditional machine learning, respectively. For *Agree*, the highest balanced accuracy was 0.691 using the paragraph-based graph representation with 17 node features; this closely matched traditional machine learning but fell short of CNN by 7.7%. Using all 17 features, with the normalisation step, provided the best performance for both *All* and *Agree*. However, for *Topical*, using only fixation duration and the number of fixations in each node without normalisation produced better results. Our assumption is that when using *All*, the model requires more information to differentiate between the different classes, but for *Topical*, a more concise view of the problem is more beneficial. Overall, the GNN approach was effective in analysing scanpaths for perceived relevance estimation for short documents in a graph classification task, outperforming the baseline approaches.

Regarding the scanpath graph representation formats, paragraph, line, and quartile-based approaches performed well with both *All* and *Agree*, with paragraph producing better balanced accuracies for both. These three representation formats might have been successful because they retained some semantic information about the text form; this requires further analysis to study the visualisation of the generated graphs superimposed over the stimuli for each classification result and see if there are indeed any noticeable patterns for each subset. However, the cluster-based graph representation using Affinity Propagation produced better balanced accuracies for *Topical*. This might be because the fixations were more focused on certain areas, e.g. certain words, so automatically generated clusters were able to find patterns unique to the respective labels, which might not have been found using paragraph, line, or quartile-based approaches; this requires further investigation to check this assumption. Graph generation is quite important and could lead to interesting research questions regarding how different approaches hold up in different applications and finding better-performing generic scanpath graph generation approaches.

5.3.2 *Node Classification Task.*

For **GoogleNQ**, the GNN approach was unable to improve the balanced accuracies for either **All** or **Agree**. The GoogleNQ corpus did not have true irrelevant paragraphs, and there was a high data imbalance between relevant and irrelevant paragraphs. Even with different graph convolutional

Table 2: The average balanced accuracy results ($\mu \pm \sigma$) for g-REL and GoogleNQ using 5-fold leave-users-out cross-validation. GNN models include Paragraph (PB), Line (LB), Cluster (CB), and Quartile-based (QB) graph structures. The subscript number indicates the total number of features (either 2 or 17), except for the LB GNN, which did not have the five area-based features. The best overall result for each corpus subset is underlined and emphasized.

		g-REL			GoogleNQ	
		All	**Agree**	**Topical**	**All**	**Agree**
Baseline Models	RF_{17}	0.624 ± 0.056	0.651 ± 0.076	0.494 ± 0.097	0.527 ± 0.064	0.484 ± 0.015
	RF^*_{17}	0.600 ± 0.045	0.650 ± 0.092	0.505 ± 0.101	0.572 ± 0.059	0.502 ± 0.108
	SVM^*_{17}	0.607 ± 0.044	0.692 ± 0.107	0.490 ± 0.124	**0.604 ± 0.047**	0.542 ± 0.103
	RF_2	0.486 ± 0.056	0.557 ± 0.139	0.444 ± 0.050	0.479 ± 0.038	0.461 ± 0.027
	RF^*_2	0.477 ± 0.049	0.570 ± 0.135	0.371 ± 0.116	0.503 ± 0.051	0.578 ± 0.218
	SVM^*_2	0.587 ± 0.065	0.692 ± 0.031	0.454 ± 0.088	0.588 ± 0.033	0.509 ± 0.106
	CNN	0.676 ± 0.078	**0.768 ± 0.107**	0.572 ± 0.086	0.552 ± 0.024	**0.603 ± 0.083**
Graph Neural Network	PB_{17}	**0.701 ± 0.021**	0.691 ± 0.114	0.486 ± 0.116	0.553 ± 0.049	0.559 ± 0.049
	LB_{12}	0.674 ± 0.050	0.668 ± 0.070	0.528 ± 0.086	–	–
	CB_{17}	0.563 ± 0.111	0.664 ± 0.073	0.621 ± 0.220	–	–
	QB_{17}	0.634 ± 0.034	0.674 ± 0.059	0.584 ± 0.129	–	–
	PB_2	0.650 ± 0.040	0.682 ± 0.103	0.591 ± 0.124	0.548 ± 0.035	0.535 ± 0.031
	LB_2	0.646 ± 0.031	0.682 ± 0.082	0.555 ± 0.081	–	–
	CB_2	0.492 ± 0.089	0.534 ± 0.031	**0.648 ± 0.205**	–	–
	QB_2	0.606 ± 0.043	0.645 ± 0.083	0.441 ± 0.132	–	–

operators, none of them produced any meaningful above chance level results. We think in order to test relevance estimation while reading longer documents that require scrolling, a more balanced dataset is required by either assigning a single label to a large document or having the same number of paragraphs corresponding to each label, in addition to having truly irrelevant paragraphs, not just relevant and topical ones. Although we cannot conclude that our approach generalises to multi-paragraph documents, the 0.604 balanced accuracy achieved by SVM* makes us believe that investigating different node classification algorithms from the literature, and using different node and edge features might lead to better results for GoogleNQ before attempting to collect a new dataset and dismissing this one.

6 CONCLUSION

This paper investigated the feasibility and potential of using GNNs for scanpath analysis for a passive gaze-based application, i.e., implicit relevance estimation during reading. Our experiments used the gazeRE dataset [1], allowing us to test GNNs for graph and node classification tasks based on texts from the g-REL and GoogleNQ corpora, respectively. We implemented a very simple GNN with three GATv2 convolutional layers. For the graph classification task, we evaluated four methods for generating graph structures from scanpaths, while for the node classification task, we used a single graph generation approach. As a baseline, we reproduced the method from Barz et al. [1] using RF and SVM classifiers with 17 eye tracking features. We also trained these classifiers using only two convex hull-based features by [2]. In addition, to compare against a neural network, we replicated the CNN approach from Bhattacharya et al. [3], which we also evaluated, for the first time, on long documents.

For **g-REL**, the GNN produced the best results for *All* and **Topical** subsets, while the CNN produced the best results for *Agree*. Based on all the presented findings, we have shown that GNNs are suitable for processing scanpath data for users' perceived relevance estimation of short text documents, which might warrant future investigation for other passive gaze-based applications. Additionally, we have shown that the CNN approach proposed by Bhattacharya et al. [3] is valid for perceived relevance estimation on short documents by evaluating it on a new dataset. For **GoogleNQ**, the node classification GNN was not successful in improving or producing meaningful above chance level results. In addition, we also could not conclude the applicability of the CNN for long documents. Based on relevant literature, we used very simple GNNs and node features in our approach to see the feasibility and potential benefits of using GNNs for scanpath processing in relevance estimation while reading. We believe the approach requires further investigation of different and more complex GNN architectures and different features, e.g., feature selection for the node features or adding edge features such as the actual distance between the nodes. Our current results suggest the feasibility of using GNNs for scanpath processing, but further studies are required to investigate its generalisability to more diverse passive gaze-based applications. In addition, future studies should consider larger datasets to validate the presented findings.

ACKNOWLEDGMENTS

This work was funded, in part, by the European Union under grant number 101093079 (MASTER), and the German Federal Ministry of Education and Research (BMBF) under grant number 01IW23002 (No-IDLE).

REFERENCES

[1] Michael Barz, Omair Shahzad Bhatti, and Daniel Sonntag. 2022. Implicit Estimation of Paragraph Relevance From Eye Movements. *Frontiers in Computer Science* 3 (2022). https://doi.org/10.3389/fcomp.2021.808507

[2] Nilavra Bhattacharya, Somnath Rakshit, and Jacek Gwizdka. 2020. Towards Real-time Webpage Relevance Prediction UsingConvex Hull Based Eye-tracking Features. In *ACM Symposium on Eye Tracking Research and Applications (ETRA '20 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3379157.3391302

[3] Nilavra Bhattacharya, Somnath Rakshit, Jacek Gwizdka, and Paul Kogut. 2020. Relevance Prediction from Eye-Movements Using Semi-Interpretable Convolutional Neural Networks. In *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval (CHIIR '20)*. Association for Computing Machinery, New York, NY, USA, 223–233. https://doi.org/10.1145/3343413.3377960 event-place: Vancouver BC, Canada.

[4] T. Blascheck, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf, and T. Ertl. 2017. Visualization of Eye Tracking Data: A Taxonomy and Survey: Visualization of Eye Tracking Data. *Computer Graphics Forum* 36, 8 (Dec. 2017), 260–284. https://doi.org/10.1111/cgf.13079

[5] Shaked Brody, Uri Alon, and Eran Yahav. 2022. How Attentive are Graph Attention Networks? https://doi.org/10.48550/arXiv.2105.14491 arXiv:2105.14491 [cs].

[6] Georg Buscher, Andreas Dengel, and Ludger van Elst. 2008. Eye Movements as Implicit Relevance Feedback. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems (CHI EA '08)*. Association for Computing Machinery, New York, NY, USA, 2991–2996. https://doi.org/10.1145/1358628.1358796 event-place: Florence, Italy.

[7] Tianle Cai, Shengjie Luo, Keyulu Xu, Di He, Tie-Yan Liu, and Liwei Wang. 2021. GraphNorm: A Principled Approach to Accelerating Graph Neural Network Training. https://doi.org/10.48550/arXiv.2009.03294 arXiv:2009.03294 [cs, math, stat].

[8] Virginio Cantoni, Chiara Galdi, Michele Nappi, Marco Porta, and Daniel Riccio. 2015. GANT: Gaze analysis technique for human identification. *Pattern Recognition* 48, 4 (April 2015), 1027–1038. https://doi.org/10.1016/j.patcog.2014.02.017

[9] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* 16 (June 2002), 321–357. https://doi.org/10.1613/jair.953

[10] Yihao Chen, Xin Tang, Xianbiao Qi, Chun-Guang Li, and Rong Xiao. 2020. Learning Graph Normalization for Graph Neural Networks. http://arxiv.org/abs/2009.11746 arXiv:2009.11746 [cs].

[11] Youngjun Cho. 2021. Rethinking Eye-blink: Assessing Task Difficulty through Physiological Representation of Spontaneous Blinking. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3411764.3445577

[12] Akbar Dehghani, Tristan Glatard, and Emad Shihab. 2019. Subject Cross Validation in Human Activity Recognition. https://doi.org/10.48550/arXiv.1904.02666 arXiv:1904.02666 [cs, stat].

[13] Andrew T. Duchowski. 2018. Gaze-based interaction: A 30 year retrospective. *Computers & Graphics* 73 (June 2018), 59–69. https://doi.org/10.1016/j.cag.2018.04.002

[14] João Marcelo Evangelista Belo, Mathias N. Lystbæk, Anna Maria Feit, Ken Pfeuffer, Peter Kán, Antti Oulasvirta, and Kaj Grønbæk. 2022. AUIT – the Adaptive User Interfaces Toolkit for Designing XR Applications. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3526113.3545651 event-place: Bend, OR, USA.

[15] Anna Maria Feit, Lukas Vordemann, Seonwook Park, Caterina Berube, and Otmar Hilliges. 2020. Detecting Relevance during Decision-Making from Eye Movements for UI Adaptation. In *ACM Symposium on Eye Tracking Research and Applications (ETRA '20 Full Papers)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3379155.3391321

[16] Brendan J. Frey and Delbert Dueck. 2007. Clustering by Passing Messages Between Data Points. *Science* 315, 5814 (Feb 2007), 972–976. https://doi.org/10.1126/science.1136800

[17] Jacek Gwizdka. 2014. Characterizing Relevance with Eye-Tracking Measures. In *Proceedings of the 5th Information Interaction in Context Symposium (IIiX '14)*. Association for Computing Machinery, New York, NY, USA, 58–67. https://doi.org/10.1145/2637002.2637011 event-place: Regensburg, Germany.

[18] Jacek Gwizdka. 2014. News Stories Relevance Effects on Eye-Movements. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '14)*. Association for Computing Machinery, New York, NY, USA, 283–286. https://doi.org/10.1145/2578153.2578198 event-place: Safety Harbor, Florida.

[19] Kenneth Holmqvist, Marcus Nystrom, Richard Andersson, Richard Dewhurst, Halszka Jarodzka, Weijer, and Joost van de. 2011. *Eye Tracking: A comprehensive guide to methods and measures.* Oxford University Press, Oxford, New York.

[20] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. https://doi.org/10.48550/arXiv.1502.03167 arXiv:1502.03167 [cs].

[21] Naji Khosravan, Haydar Celik, Baris Turkbey, Elizabeth C. Jones, Bradford Wood, and Ulas Bagci. 2019. A collaborative computer aided diagnosis (C-CAD) system with eye-tracking, sparse attentional model, and deep learning. *Medical Image Analysis* 51 (Jan. 2019), 101–115. https://doi.org/10.1016/j.media.2018.10.010

[22] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. https://doi.org/10.48550/arXiv.1412.6980 arXiv:1412.6980 [cs].

[23] Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. (2016). https://doi.org/10.48550/ARXIV.1609.02907

[24] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics* 7 (Aug. 2019), 453–466. https://doi.org/10.1162/tacl_a_00276

[25] Guohao Lan, Bailey Heit, Tim Scargill, and Maria Gorlatova. 2020. GazeGraph: graph-based few-shot cognitive context sensing from human visual behavior. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems (SenSys '20)*. Association for Computing Machinery, New York, NY, USA, 422–435. https://doi.org/10.1145/3384419.3430774

[26] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. 2017. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research* 18, 17 (2017), 1–5. http://jmlr.org/papers/v18/16-365.html

[27] Beibin Li, Nicholas Nuechterlein, Erin Barney, Claire Foster, Minah Kim, Monique Mahony, Adham Atyabi, Li Feng, Quan Wang, Pamela Ventola, Linda Shapiro, and Frederick Shic. 2021. Learning Oculomotor Behaviors from Scanpath. In *Proceedings of the 2021 International Conference on Multimodal Interaction*. ACM, Montréal QC Canada, 407–415. https://doi.org/10.1145/3462244.3479923

[28] Xiaochuan Ma, Yikang Liu, Roy Clariana, Chanyuan Gu, and Ping Li. 2023. From eye movements to scanpath networks: A method for studying individual differences in expository text reading. *Behavior Research Methods* 55, 2 (Feb. 2023), 730–750. https://doi.org/10.3758/s13428-022-01842-3

[29] Päivi Majaranta and Andreas Bulling. 2014. *Eye Tracking and Eye-Based Human–Computer Interaction.* Springer, London, 39–65. https://doi.org/10.1007/978-1-4471-6392-3_3

[30] Abdulrahman Mohamed Selim, Michael Barz, Omair Shahzad Bhatti, Hasan Md Tusfiqur Alam, and Daniel Sonntag. 2024. A review of machine learning in scanpath analysis for passive gaze-based interaction. *Frontiers in Artificial Intelligence* 7 (June 2024). https://doi.org/10.3389/frai.2024.1391745 Publisher: Frontiers.

[31] Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. 2018. Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks. (2018). https://doi.org/10.48550/ARXIV.1810.02244

[32] Erik Novak, Luka Bizjak, Dunja Mladenić, and Marko Grobelnik. 2022. Why is a document relevant? Understanding the relevance scores in cross-lingual document retrieval. *Knowledge-Based Systems* 244 (2022), 108545. https://doi.org/10.1016/j.knosys.2022.108545

[33] Douglas Oard and Jinmook Kim. 1998. Implicit Feedback for Recommender System. *Proceedings of the AAAI Workshop on Recommender Systems* (1998). https://cs.fit.edu/~pkc/apweb/related/oard-aaaiWS98.pdf

[34] Pernilla Qvarfordt. 2017. Gaze-informed multimodal interaction. In *The Handbook of Multimodal-Multisensor Interfaces: Foundations, User Modeling, and Common Modality Combinations - Volume 1*, Sharon Oviatt, Björn Schuller, Philip R. Cohen, Daniel Sonntag, Gerasimos Potamianos, and Antonio Krüger (Eds.). ACM, 365–402. https://doi.org/10.1145/3015783.3015794

[35] Lei Shi, Cosmin Copot, and Steve Vanlanduit. 2021. Gaze Gesture Recognition by Graph Convolutional Networks. *Frontiers in Robotics and AI* 8 (2021). https://doi.org/10.3389/frobt.2021.709952

[36] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. https://doi.org/10.48550/arXiv.1409.1556 arXiv:1409.1556 [cs].

[37] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph Attention Networks. (2017). https://doi.org/10.48550/ARXIV.1710.10903

[38] Bin Wang, Hongyi Pan, Armstrong Aboah, Zheyuan Zhang, Elif Keles, Drew Torigian, Baris Turkbey, Elizabeth Krupinski, Jayaram Udupa, and Ulas Bagci. 2024. GazeGNN: A Gaze-Guided Graph Neural Network for Chest X-Ray Classification. 2194–2203. https://openaccess.thecvf.com/content/WACV2024/html/Wang_GazeGNN_A_Gaze-Guided_Graph_Neural_Network_for_Chest_X-Ray_Classification_WACV_2024_paper.html

[39] Ryen W. White, Ian Ruthven, and Joemon M. Jose. 2002. The Use of Implicit Evidence for Relevance Feedback in Web Retrieval. In *Advances in Information*

Retrieval (Lecture Notes in Computer Science), Fabio Crestani, Mark Girolami, and Cornelis Joost van Rijsbergen (Eds.). Springer, Berlin, Heidelberg, 93–109. https://doi.org/10.1007/3-540-45886-7_7

[40] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 1 (Jan. 2021), 4–24. https://doi.org/10.1109/TNNLS.2020.2978386

[41] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *International Conference on Learning Representations*. https://openreview.net/forum?id=ryGs6iA5Km

[42] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. 1996. BIRCH: an efficient data clustering method for very large databases. *ACM SIGMOD Record* 25, 2 (Jun 1996), 103–114. https://doi.org/10.1145/235968.233324

[43] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open* 1 (Jan. 2020), 57–81. https://doi.org/10.1016/j.aiopen.2021.01.001