Unknown Word Detection for English as a Second Language (ESL) Learners using Gaze and Pre-trained Language Models

Jiexin Ding
Department of Computer Science and Technology, Global Innovation Exchange (GIX) Institute
Tsinghua University
Bejing, China
Electrical & Computer Engineering
University of Washington
Seattle, Washington, USA
jxding17@gmail.com

Bowen Zhao
Groundlight AI
Seattle, Washington, USA
bowen@groundlight.ai

Yuntao Wang*
Department of Computer Science and Technology
Tsinghua University
Beijing, China
yuntaowang@tsinghua.edu.cn

Xinyun Liu
Rice University
Houston, Texas, USA
liuxinyun6@yahoo.com

Rui Hao
School of Artificial Intelligence
University of Chinese Academy of Sciences
Beijing, China
haorui24@mails.ucas.ac.cn

Ishan Chatterjee
Paul G. Allen School of Computer Science & Engineering
University of Washington
Seattle, Washington, USA
ichat@cs.washington.edu

Yuanchun Shi
Department of Computer Science and Technology
Tsinghua University
Beijing, China
Qinghai University
Xining, Qinghai, China
shiyc@tsinghua.edu.cn

Abstract

English as a Second Language (ESL) learners often encounter unknown words that hinder their text comprehension. Automatically detecting these words as users read can enable computing systems to provide just-in-time definitions, synonyms, or contextual explanations, thereby helping users learn vocabulary in a natural and seamless manner. This paper presents EyeLingo, a transformer-based machine learning method that predicts the probability of unknown words based on text content and eye gaze trajectory in real time with high accuracy. A 20-participant user study revealed that our method can achieve an accuracy of 97.6%, and an F1-score of 71.1%. We implemented a real-time reading assistance prototype to show the effectiveness of EyeLingo. The user study shows improvement in willingness to use and usefulness compared to baseline methods.

CCS Concepts

• **Human-centered computing → Interaction techniques**.

Keywords

Unknown word detection, gaze, pre-trained language model.

ACM Reference Format:
Jiexin Ding, Bowen Zhao, Yuntao Wang, Xinyun Liu, Rui Hao, Ishan Chatterjee, and Yuanchun Shi. 2025. Unknown Word Detection for English as a Second Language (ESL) Learners using Gaze and Pre-trained Language Models. In *CHI Conference on Human Factors in Computing Systems (CHI '25), April 26–May 01, 2025, Yokohama, Japan.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3706598.3714181

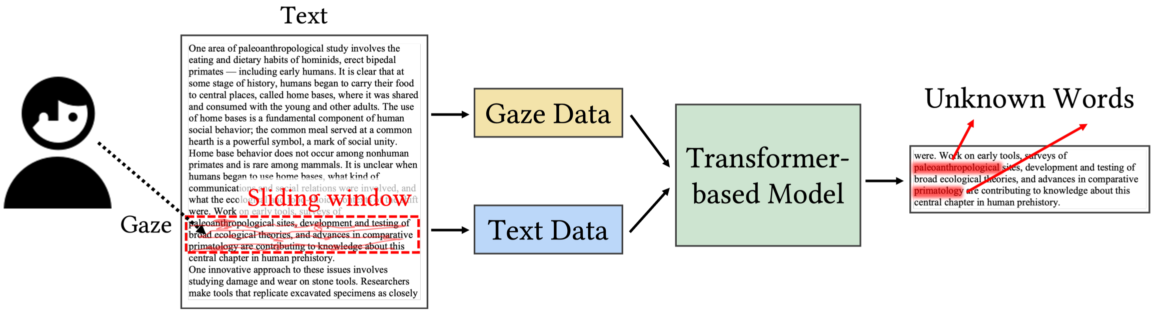
1 Introduction

Unknown words can greatly reduce reading fluency and worsen the reading experience of non-native speakers [31, 37]. By automatically detecting these unknown words as users read, computing systems can assist users in reading and language comprehension, and provide just-in-time word explanations for learning vocabulary. Because unknown words vary among users, many previous works were based on the explicit expression of intention by users, such as mouse clicks [15] or looking at words intentionally [7, 14]. To facilitate a more natural reading experience, other methods

*denotes as the corresponding author.

CHI '25, Yokohama, Japan

ACM ISBN 979-8-4007-1394-1/25/04
https://doi.org/10.1145/3706598.3714181



One area of paleoanthropological study involves the eating and dietary habits of hominids, erect bipedal primates — including early humans. It is clear that at some stage of history, humans began to carry their food to central places, called home bases, where it was shared and consumed with the young and other adults. The use of home bases is a fundamental component of human social behavior; the common meal served at a common hearth is a powerful symbol, a mark of social unity. Home base behavior does not occur among nonhuman primates and is rare among mammals. It is unclear when humans began to use home bases, what kind of communications and social relations were involved, and what the ecological and [illegible] contexts of the shift were. Work on early tools, surveys of paleoanthropological sites, development and testing of broad ecological theories, and advances in comparative primatology are contributing to knowledge about this central chapter in human prehistory.
One innovative approach to these issues involves studying damage and wear on stone tools. Researchers make tools that replicate excavated specimens as closely

were. Work on early tools, surveys of paleoanthropological sites, development and testing of broad ecological theories, and advances in comparative primatology are contributing to knowledge about this central chapter in human prehistory.

Figure 1: Our method locates the content the user is reading in real-time through gaze, and inputs the gaze data and text data to the transform-based model to detect unknown words.

use gaze characteristics such as fixation duration, number of fixations, and saccade length [17, 41] to detect unknown words automatically, since there is a correlation between gaze and word difficulty [25]. Previous research shows that fixation, the maintaining of the gaze on a single location, happens when people are processing phrases [25]. However, these gaze-based methods have two major challenges affecting their accuracy and ease of deployment. First, these methods require dedicated and costly eye-tracking hardware to obtain accurate eye movement data for these features. Moreover, even with professional eye trackers, measuring gaze is inherently inaccurate due to complicated eye motions, making it hard to precisely map a gaze point to the word in the text being read [3]. To reduce reliance on gaze information, other works seek to compensate or replace inaccurate eye-tracking data on commercial devices by incorporating text [12, 16, 20, 24], click [15] and motion data [19].

Besides the cognitive process reflected by gaze, inherent linguistic information about words is also crucial for identifying unknown words. With the development of Natural Language Processing (NLP) technology, pre-trained language models (PLMs) demonstrate a powerful ability to capture rich linguistic information [11, 29] which is strongly related to word difficulty [18]. We explore how to take advantage of the language model in addition to gaze to make unknown word detection accurate, easily accessible, and more applicable. We present a real-time unknown word detection method that locates a region of interest based on gaze and then integrates linguistic information provided by PLM and gaze trajectory to predict unknown words using a transformer-based model. In this way the inaccuracy of the gaze-based method can be compensated by the probabilities distributed on words derived from the language model.

As shown in Fig. 1, we tackle the problem of unknown word detection with two modalities of information, gaze and text. To process gaze patterns, we utilize the cutting-edge transformer-based encoder-decoder model with cross-attention modules to learn the positional information based on gaze trajectory and text layout. For linguistic information, we apply RoBERTa to the text in the region of interest with several crucial word-level knowledge embeddings (term frequency, part-of-speech, etc.). By jointly training the models above on our collected dataset, our approach surpasses existing methods with 71.1% F1-scores and 97.6% accuracy for unknown word detection. With our method, real-time language learning assistance and just-in-time vocabulary acquisition tools can be enabled.

In this paper, we first verified the feasibility of our approach on the dataset collected by a professional eye tracker. We also applied our unknown word detection method to the relatively noisy webcam-based gaze data to show the robustness of our method with commodity deployments. Then, we conducted experiments to analyze the contribution of gaze and PLM. Finally, we implemented a reading assistance prototype to evaluate our method in a real world setting where data is processed in real time and investigate its usefulness and users' subjective experience.

Contributions in this work are summarized as follows:

(1) We propose an unknown word detection method (EyeLingo) that leverages gaze to locate a region of interest and then classifies wrods based on linguistic characteristics provided by a pre-trained language model and gaze encoding derived from a transformer-based model. It achieves an accuracy of 97.6% and an F1-score of 71.1%.
(2) We analyze the contributions of gaze and the pre-trained language model. Unknown word detection is mainly based on linguistic characteristics provided by a pre-trained language model. Gaze provides user-dependent and timely information to detect different unknown words for different users in a real-time manner.
(3) We build a reading assistance prototype and conduct a real-time evaluation. The result shows that the F1-score is 56.5%. The user study shows improvement in willingness to use and usefulness compared to the traditional method.

2 Related Work

In this section, we illustrate how gaze behavior is related to cognitive processes in reading and examine the reading behavior analysis enabled by gaze tracking. Then, we explain the limitations of the gaze-based unknown word detection method and how the previous works improved their performance. After that, we analyze the gaps in detecting unknown words accurately.

2.1 Gaze in Reading

Reading as a cognitive process affects eye movement [25]. When people are reading, the eyes follow the text through small amplitudes, ballistic motions called saccades. The pauses between two saccades are called a fixation [24]. According to the eye-mind hypothesis, the fixation on a word persists during its processing phase [25]. Consequently, fixation duration can be a metric for identifying difficult words and measuring cognitive processes in reading. For this reason, previous research combines fixation duration with other gaze features to predict reading comprehension [8, 33, 39], detecting mind wandering [6], identifying interest [13], and detecting attention [21, 26, 46] in reading. Other works facilitate natural language processing tasks such as named entity recognition and sequence classification by integrating gaze into neural network [1, 2, 22, 30]. Besides, some researches also improve the prediction of gaze behavior such as scanpath and calibration process based on the connection between gaze and text [10, 27, 42]. However, most of these works focus on paragraph-level and sentence-level tasks rather than word detection.

Extended periods of fixation on the focal word suggest difficulties in word identification [24], forming the theoretical basis for the detection of unknown words via gaze. However, it is hard to achieve high accuracy only based on the fixation because of the inaccuracy of gaze-tracking hardware, algorithms, and the ambiguous relationship between gaze patterns and the cognitive processes of understanding words. The highest accuracy of eye tracking is around 0.3° (2.6 mm when the distance is 50 cm) under optimal conditions, but the accuracy can easily be affected by the calibration performance and user posture [28]. Thus, how to accurately match the gaze point to the focused text is a problem. Additionally, other researchers pointed out that the processing of words can occur when they are not held in fixation [36].

Although there is a strong correlation between gaze behavior and word difficulty, the above limitations make it difficult to achieve high accuracy in detecting unknown words based only on gaze. Previous gaze-based unknown word detection methods improve their accuracy by combining multiple gaze features and leveraging text information such as word length and word rarity [16, 20].

2.2 Unknown Word Detection

According to the importance of gaze in analyzing reading behavior as explained above, most of the related works are gaze-based. iDict [24] detects unknown words based on gaze duration and word frequency and sets a threshold to trigger a gloss or margin note on unknown words. It successfully detects 36.5% of unknown words. Later works extend iDict by replacing the threshold function with machine learning. Hiroka et al. [20] uses several gaze features such as first gaze duration, number of fixations, and number of regressions and feeds them into support vector machines (SVM) to classify the unknown word. Their model performs best (F1-score of 55.6%) when adding linguistics features including word length and word rarity. Similarly, Garain et al. [16] also take both gaze and linguistics features into consideration to achieve the best F1-score of 86% on a single user using an SVM. However, each of these methods rely on a professional eye tracker to obtain sufficiently accurate gaze data to extract gaze features and match these features to the words. Even with a dedicated eye tracker, it is still difficult to accurately assign each gaze point to the corresponding line due to the vertical error [38, 45]. Therefore, in their experiment, the line spacing was set between 3.0 to 6.0 to better distinguish lines based on the y-value. The large line spacing makes it difficult for their method to be applied in real-world scenarios, considering that the line spacing of text we usually read is mostly between 1.0 and 2.0. Moreover, their data preprocessing that relies on global gaze coordination makes them even harder to be applied in a real-time application.

Apart from gaze, other reading behaviors such as mouse clicks and hand motion can be used to detect unknown words. Ehara et al. [15] gives users potential highlights in advance and analyzes users' feedback based on their clicks on the web page. Predictions are improved based on this feedback, reaching an accuracy of up to 80.01%. Higashimura et al. [19] targets at vocabulary acquisition on smartphones and identifies unknown words utilizing the motion data obtained from the inertia sensors on smartphones. The estimation improves through the reading and the AUPR is about 0.3. Both methods are device-specific and only applicable to a single user as they need to be optimized based on personalized iterative feedback for higher accuracy.

In summary, the current best-performing detection methods combine gaze and text data. However, these current methods still rely on heavy gaze data preprocessing and special experimental settings. A real-time method that is robust to relatively inaccurate gaze data can promote the availability of unknown-word detection. Since word difficulty is highly dependent on linguistic features[18], we seek to build upon the rapid development of natural language processing (NLP) technology [43] in recent years to yield a solution. Pre-trained language models such as [11, 29] encompass extensive text information for downstream applications. We harness their capabilities in a new architecture to increase the tolerance of inaccuracy in gaze tracking. We propose a transformer-based model that requires linguistic information provided by PLM from a region of interest around the target words and learns the gaze pattern automatically from gaze trajectory and text position.

3 Unknown Word Detection Method

In this section, we explain why we chose gaze and PLM to detect unknown words. Then we describe how we collected and processed the data. Finally, we explain the architecture of our method in detail.

3.1 Method Justification

Even though whether a word is unknown to a user is subjective, linguistic priors related to word difficulty are helpful in unknown word prediction. Therefore, we assume a user's unknown word can be predicted based on (1) linguistic characteristics related to word difficulty, such as frequency, contextual distinctiveness, and COCA (Corpus of Contemporary American English) range, and (2) user-dependent factors which can be reflected by gaze trajectory. Therefore, we integrate text and gaze information to detect unknown words as shown in Fig. 2:

(1) Linguistic characteristics of the target word's tokens and the textual context in the region of interest are captured by RoBERTa. RoBERTa is an effective and efficient language model for natural language understanding, ideally fitting our

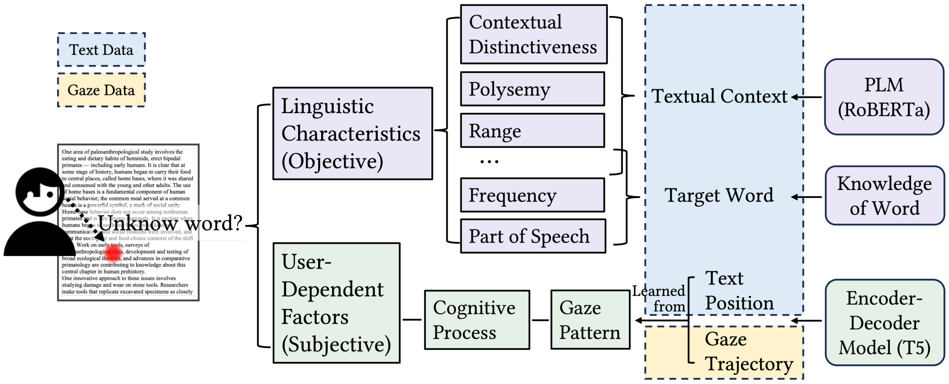


Figure 2: Methodology for Detecting Unknown Words Using Integrated Text and Gaze Information. This integrated approach leverages both linguistic and user-dependent factors to effectively identify unknown words.

expectations, based on its proven performance in capturing complex linguistic features, such as contextual embeddings, syntactic structures, and semantic relationships. Prior research demonstrated that models like RoBERTa outperform traditional methods in tasks requiring a deep understanding of natural language [29]. Considering that the context around the word influences word difficulty [18], we applied RoBERTa to all the text in the region of interest. Besides, as we aim at deploying EyeLingo to end-user devices like laptops, we do not consider using billion-level large language models for efficiency reasons. Meanwhile, generative models are also unsuitable in our use case, as our task is to identify users' unknown words in the given text rather than generating new content. Therefore, we choose RoBERTa instead of GPT. The model details related to this are explained in Section 3.2.2.

(2) Word-level knowledge of the target word such as term frequency, part of speech and named entity recognition are applied to enhance the performance. The tokenization is required to use RoBERTa. The word-level linguistic characteristics are partially lost during tokenization. Thus, we introduce word-level knowledge to compensate for the loss. The model details related to this are explained in Section 3.2.3.

(3) Gaze pattern is automatically learned from gaze trajectory and text position by the model adapted from T5, one of the top encoder-decoder transformer-based architectures. Gaze patterns can reflect a user's cognitive process, indicating whether the user encounters an unknown word [25]. Considering the complexity of gaze patterns and the variability between users, we want to automatically learn these patterns using machine learning models based on gaze trajectory and text position. The encoder-decoder model can learn complex relationships between different types of sequential data. T5 has demonstrated cutting-edge performance on a wide range of tasks and is capable of capturing complex patterns and dependencies within input sequences [34]. We take out the output of the last encoder layer as the positional encoding which represents the relationship between the gaze trajectory and text position. The model details related to this are explained in Section 3.2.1.

Our method has three main advantages compared to previous works:

(1) EyeLingo has a higher tolerance for the noise of gaze data and has better performance by leveraging language models' ability to capture linguistic characteristics to reduce the reliance on gaze (results in Section 4.2 and Section 4.4).

(2) EyeLingo eliminates tedious feature engineering by automatically learning gaze patterns based on gaze trajectory and text coordinates (detailed explanations in Section 3.2.1 and results in Section 4.3).

(3) EyeLingo enables real-time detection by only relying on the local information in the region of interest, which is benefit from the low reliance on gaze (results in Section 5).

3.2 Unknown Word Detection Model

The goal of our model is to classify whether a word is an unknown word or not using both gaze information and text information. We use the encoder-decoder architecture to encode positional gaze and text information into the vector space. We also leverage RoBERTa [29] to integrate with text information. Moreover, we also add the prior knowledge to take the use of the word-level information. The overall architecture of our model is shown in Fig. 3.

3.2.1 Positional Data Encoding. To accurately capture the correlation between the user's gaze and the document, we adapt the model architecture from T5 [34], the state-of-the-art encoder-decoder language model to process the positional data from those two modalities, where the encoder learns to capture the user's gaze pattern and the decoder is expected to predict whether a token belongs to an unknown word of the user, based on the encoder's outputs and the tokens' positional data. For the encoder part, we feed the raw and the moving averaged gaze trace to let the model capture both the fine-grained and general positional information from the user's behavior. As for the decoder part, instead of using the token-level positional data as the inputs only, we also calculate the

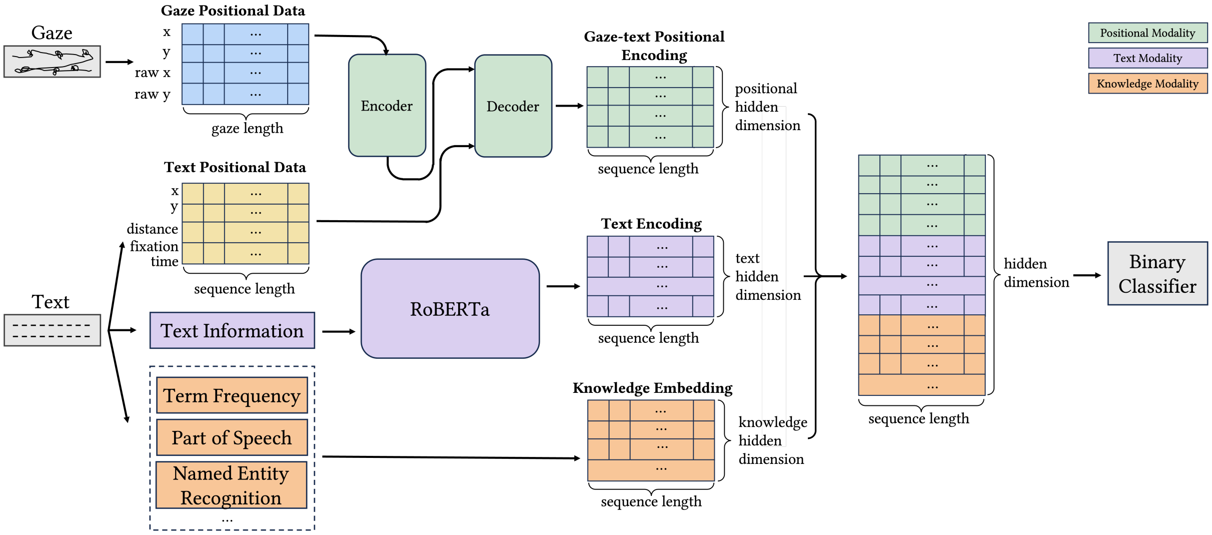


Figure 3: Our model includes an encoder-decoder model to encode positional data, a pre-trained RoBERTa to encode text information, and learnable embeddings to encode the knowledge. The concatenation of these three matrices is input to a binary classifier.

averaged gaze-token distance $d(g,w)$ and the gaze duration $t(g,w)$ (the length of the time when the user's gaze lives in the bounding box of the token) for each token w

$$d(g,w)=\sqrt{\left(\frac{\sum_{i=1}^{N_g} g_x^i}{N_g}-\frac{w_x^s+w_x^t}{2}\right)^2+\left(\frac{\sum_{i=1}^{N_g} g_y^i}{N_g}-\frac{w_y^s+w_y^t}{2}\right)^2} \tag{1}$$

$$t(g,w)=|\{(g_x^i,g_y^i)|1\le i\le N_g \wedge w_x^s\le g_x^i\le w_x^t \wedge w_y^s\le g_y^i\le w_y^t\}| \tag{2}$$

where N_g is the number of gaze samples within the sliding window, $w_x^s, w_x^t, w_y^s, w_y^t$ are the coordinates of the bounding box of the token w, and g_x and g_y are the user's gaze trace on x and y axis, respectively. Overall, this encoder-decoder model can be summarized as

$$H_g=\text{Encoder}(g_x,g_y,g_x^{raw},g_y^{raw}) \tag{3}$$

$$P=\text{Decoder}(H_g,w_x,w_y,d(g,w),t(g,w)) \tag{4}$$

where g_x, g_y are the moving-averaged gaze positional data, g_x^{raw}, g_y^{raw} are the raw gaze data without filters applied, w_x, w_y are the positions of each token, and H_g is the final encoder outputs, which is used as the inputs (more specifically, keys and values) of the cross-attention modules in decoder layers.

3.2.2 Textual Information Capturing. We utilize a pre-trained RoBERTa, a widely used pre-trained language model based on the transformer architecture, to encode the text information. The layer consists of a self-attention module and a feed-forward layer. The structure can help the model better encode the text by using the surrounding text to establish the context. It encodes text data $s \in \mathbb{R}^{n_{txt}}$ to $Z \in \mathbb{R}^{n_{txt}\times n_r}$, in which n_r is the hidden dimension of RoBERTa.

$$Z=\text{RoBERTa}(s) \tag{5}$$

3.2.3 Knowledge-grounded Enhancement. The pre-trained language model takes tokens instead of words as input. Word-level information may be lost during tokenization. Therefore, we introduce word-level knowledge including the term frequencies, part of speech [4], and named entity recognition [23] to utilize word-level information. These features are encoded into a knowledge matrix $K \in \mathbb{R}^{n_{txt}\times n_k}$.

3.2.4 Training. We combine the encodings of the three modules' outputs as the inputs for the final classifier layer, a logistic regression module for binary classification. We fully fine-tune our model, including the positional encoder-decoder module, the RoBERTa pre-trained module, the knowledge embeddings, along with the final classifier on our dataset. It shall be noticed that the task of unknown word detection is significantly class imbalanced, where on average there are 15 times more negative tokens (known words) than positive ones (unknown words). To mitigate this issue, we use the focal binary entropy loss for our model's training which re-weights the loss term for more robust training:

$$H=[P;Z;K] \tag{6}$$

$$p=\sigma(W_o\cdot H+b_o) \tag{7}$$

$$\mathcal{L}(p,\hat{y})=-\alpha\hat{y}(1-p)^{\gamma}\log(p)-(1-\alpha)(1-\hat{y})p^{\gamma}\log(1-p) \tag{8}$$

where p is the model's prediction logits, $\hat{y}$ is the ground truth, while α and γ are the hyper-parameters that control the weight between the two classes and the speed of the model's focus on difficult examples, respectively.

3.3 Data Preparation

3.3.1 Implementation. We built a system to collect gaze data from eye trackers and webcams at the same time. We used Tobii Pro Nano[1] eye tracker whose sampling rate is 60Hz and streamed its

[1] https://www.tobii.com/products/eye-trackers/screen-based/tobii-pro-nano

data to the computer through a Python script. The accuracy of the optimal conditions of Tobii Nano eye tracker is 0.3°. For webcam data, we used a SeeSo[2], a remote eye tracking platform, by integrating it into our PDF reader. To get text information including their contents and positions, we built a web-based PDF reader based on an open source Github repository[3]. This platform can record gaze data using Seeso, retrieve text information when users finish reading, and record users' clicks while they are labeling their unknown words after the reading.

The laptops we used were Macbook Pro (CPU: Apple M1 Pro, RAM: 16 GB, Storage: 512 GB, Screen Size: 14.2 inches, Resolution: 1512×982)[4] and Huawei MateBook D14 2022 (CPU: i5-1155G7, RAM: 16 GB, Storage: 512 GB, Screen Size: 14 inches, Resolution: 1920 × 1080)[5]. The height of the line is 20.63 pixels on Macbook and 20.74 on Matebook, so the height of each line of text is approximately 4.1 mm on the MacBook and about 3.3 mm on the MateBook during the experiment. According to the Tobii calibration requirement[6], the distance between the participant's face and the screen during the experiment is about 50 − 60 cm. Thus, the 10-point font size can be converted to 0.39° − 0.47° on a MacBook and 0.32° − 0.39° on a MateBook.

3.3.2 Participant and Material. We recruited a total of 20 undergraduate and graduate students (5 females, 15 males) whose second language is English. Their Vocabulary Levels Test (VLT) scores ranged from 7 to 30 ($M = 20.24, SD = 7.15$). The VLT is widely used to measure the vocabulary level of English learners [32, 40]. We chose the 5000-word frequency level group because the TOEFL reading materials we used for the user study presume a vocabulary knowledge of approximately 4500 words [9]. Their ages range from 21 to 26 years old ($M = 22.85$, $SD = 1.65$). Among them, 16 people wore glasses during data collection and 4 did not. The reading materials contain 120 articles of TOEFL and GRE reading with an average length of 363 words per article and 43534 words in total. We organized the text into a common paper format, which is single-spaced and has two columns. The font is Times New Roman and the font size is 10.

3.3.3 Experiment Design and Procedure. To increase text diversity, we divided twenty participants into four groups, and participants in each group read the same 30 articles. These 30 articles were divided into 3 days to read, which took about an hour each day. There was a calibration session before data collection for each participant each day to calibrate both the eye tracker and webcam eye tracker. Participants can take a break whenever they feel fatigued. Typically, participants take a break after reading 3-5 articles. If they chose to take a break, there would be a re-calibration before the collection was restarted, considering that the person's sitting posture had a great impact on the accuracy of eye tracking. The average reading time per article is 2.88 minutes ($SD = 1.01$), which means the continuous reading time ranges from 8 to 14 minutes after one calibration.

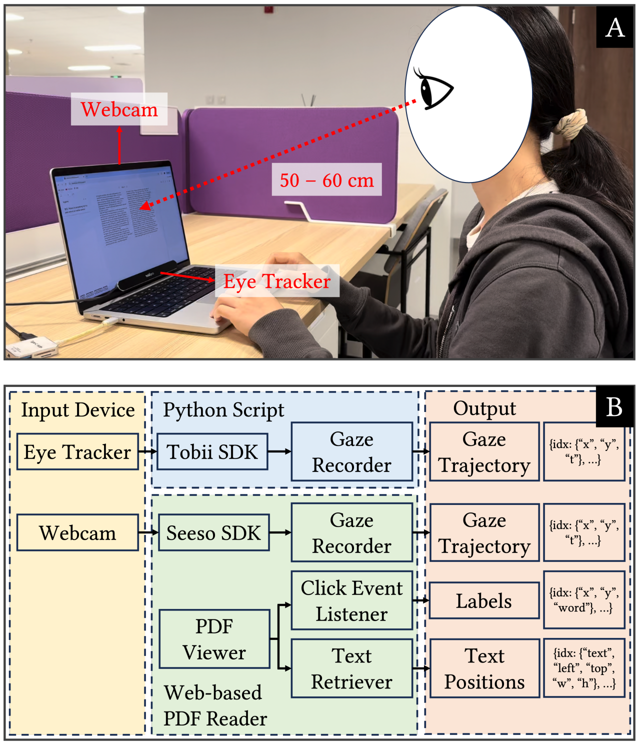


Figure 4: (A) During data collection, users are seated at a distance of approximately 50-60 cm from the screen, as instructed by the Tobii calibration interface. The eye tracker and webcam simultaneously collect data. (B) The data collection platform includes a Python script to read eye tracker data and a web-based PDF reader to read webcam data and record word labeling.

Following the previous work [16], we separated the collection of gaze data and the labeling of unknown words in order to avoid mouse clicks from affecting the user's normal eye movement behavior when reading. Participants read each article twice. During the first pass, eye movement data were collected while participants were reading. In the second pass, participants were asked to mark the unknown words they encountered in the first pass, and eye movement data were not collected for the second pass. Participants started the second pass right after the first pass to restore the feelings of the first time and help participants recall as much as possible the unknown words they encountered in the first pass. In addition, before participants read each article for the first time, the experimenter clicked a button to download all the words on the current page along with their corresponding coordinates. During the second pass, participants were required not to zoom or move the webpage to ensure that the positions of the text remained unchanged.

3.3.4 Data Preprocessing. To support real-time unknown word detection, we used a sliding time window to segment the data. The length of the window is 1 second, and the overlap between the two windows is zero. The data processing includes two steps.

[2] https://seeso.io/
[3] https://github.com/zotero/pdf-reader
[4] https://support.apple.com/en-us/111902
[5] https://consumer.huawei.com/en/laptops/matebook-d-14-2022/specs/
[6] https://connect.tobii.com/s/article/how-to-position-participants-and-the-eye-tracker?language=en_US

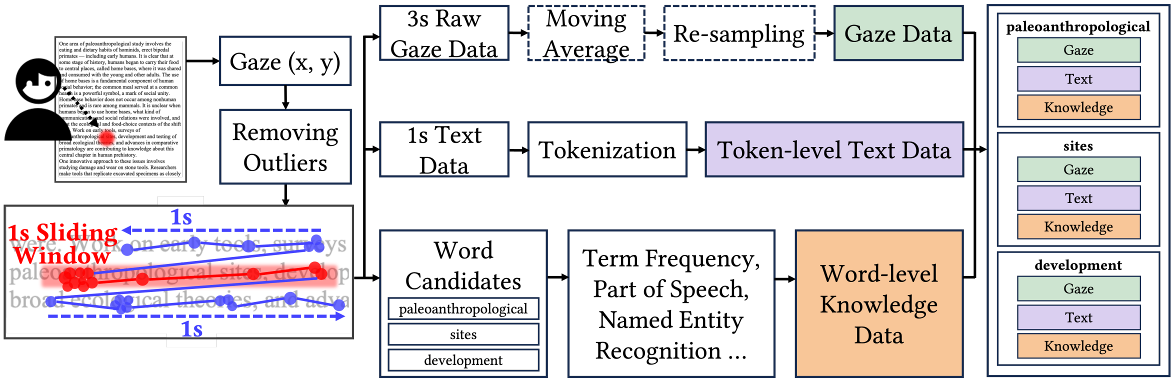


Figure 5: A bounding box is derived from the gaze coordination within a 1-second sliding window. The gaze data, token-level text data, and word-level knowledge data are calculated for each candidate word in the bounding box.

First, we located the text the user read within 1 second using the coordination of the gaze sequence. Then, we processed the raw gaze data and text data into 3 types of information which are gaze, text, and knowledge for each word in the sliding window.

As shown in the left part of Fig. 5, we firstly de-noised the gaze data to locate the content the user is reading by getting the bounding box of the gaze coordination. Blinking can cause sudden changes in gaze data in the y direction, which will cause the bounding box to be abnormally large. We removed these outliers to avoid the extra-large bounding boxes. We analyzed the distribution of the range of the y-coordinates of the gaze data within 1 second. When the distance between a small subset of the data in a window and the other part of the data exceeds the width of three lines, we removed the smaller portion of the data. If the y-coordinates of all data in the window fluctuated greatly, we ignored this window.

After getting the bounding box of 1s gaze (red dotted box in Fig. 5), all the words except the function word (such as articles and conjunctions) in the box were regarded as the word candidate. For gaze data, the 1-second window was extended to the 3-second window by adding one second before and after the window. The consideration for this extension is that the word identification span is about 7-8 characters [35] and the average reading speed of participants is 2.51 words per second. We applied the moving average filter and re-sampled the gaze data collected by webcam but not eye tracker, because webcam-based data is more noisy and the sampling rate of it varies from 24Hz to 27Hz. For text data, we tokenized the content within the 1-second window. We also added prior knowledge such as the term frequencies, part of speech, and named entity recognition to each candidate word. In the example shown in Fig. 5, finally we got 3 samples from this 1-second window which are 3 words with their gaze data, token-level text data, and word-level knowledge. Labels of these words are directly derived from the mouse click file.

Compared with the data collected with eye trackers, the data collected with the webcam is noisier, and the data quality is more affected by the user's sitting posture. Therefore, we aligned the data of the first article read by the user in each session with the coordinates of the article and applied the parameters of this alignment to other articles in this session.

4 Experiment

In this section, we systematically evaluated our model. First, we present the results of our model on eye tracker data and webcam data. Second, we analyzed the contributions of gaze and the pre-trained language model (PLM) separately. Finally, we discuss several important aspects for practical applications, including the model's generalizability, latency, and memory consumption.

4.1 Experiment Settings

4.1.1 Training and Evaluation of EyeLingo. The statistics of our collected dataset are shown in Table 1. In the default setting (main model), the data from all users and all articles are mixed and divided into training set, development set and testing set according to 8:1:1. We fully fine-tuned our model for 30 epochs with a batch size of 32 on our dataset. The learning rate of the positional encoder-decoder model is set as 1e-3, while the rest of the model's learning rate, including the RoBERTa backbone, knowledge embeddings, and the classifier are set as 2e-5. The sample rate of the user's gaze is set as 60 Hz, with a maximum length of 3 seconds, while the maximum number of context tokens is set as 64.

We report word-level accuracy, precision, recall, and F1-score in our experiments. In detail, a word is recognized as unknown if there exists at least one token within it predicted as positive. We report the F1 score on the test dataset, after selecting the best model based on the F1 score on the dev set throughout training with early stopping, with the binary classification threshold searched between 0 to 1 with the step of 0.01. The result is shown in Table 2.

4.1.2 Baseline. We compared our method to an SVM baseline implemented according to [16] and three simple baselines. For distance and fixation heuristics baseline, the distance and fixation time calculation follow those described in Section 3.2.1. The only difference is that we use the bounding box of words instead of tokens in this case. The logistic regression model is trained with these two features above, plus word term frequency, part of speech, and named entity tags. Results are shown in Table 2.

When implementing the SVM baseline, we used four gaze features (number of forward gaze points, number of backward gaze

Table 1: Statistics of our collected data.

#Users	#Documents	#Train Data	#Dev Data	#Test Data	#Tokens	#Unknown Tokens
20	120	9,802	980	980	380,524	25,233

Table 2: The Main Results of our method compared to heuristic and logistic regression baselines, backed with eye tracker and webcam collected user gaze data. Our model on the eye tracker data performs the best (highlighted in bold).

Device	Method	Accuracy (%)	F1 (%)	Precision (%)	Recall (%)
Eye tracker	Distance heuristics	76.6	20.6	13.3	45.5
Eye tracker	Fixation heuristics	82.4	22.9	16.2	39.1
Eye tracker	Logistic regression	96.6	23.4	15.9	44.5
Eye tracker	SVM [16]	83.0	29.9	20.3	56.7
Eye tracker	Ours (main model)	**97.6**	**71.1**	**63.3**	**79.0**
Webcam	Ours (main model)	97.3	65.1	60.3	69.7

points, duration of forward reading, duration of backward reading) plus two linguistic features (word length, word frequency) mentioned in [16]. Because the articles we chose are longer than those in previous work, there were lots of rereads, making it difficult to determine whether it is a new line or not based on sudden changes in the x value. In addition, due to the smaller line spacing, correcting the line number based on the y value is impossible. Therefore, we only assigned line numbers to the gaze based on which line was closest to the gaze coordination and used a median filter to remove outliers. When choosing the parameters for SVM, we did the grid search and used the best one for the evaluation. The result we reported was trained using SVM (RBF kernel, C=0.01, gamma=0.1) with a class weight ratio of 6:1 between positive and negative classes.

4.1.3 Ablation Study. As shown in Table 3, we conducted ablation studies on our method by removing the contextual encoding (the use of PLMs), gaze encoding (the positional encoder-decoder), and the knowledge embeddings, respectively. When ablating the text encoding component (the PLM), we remove the text information in the inputs and discard the text encoding in our model. The model should predict the users' unknown words based on their gaze trajectories' relationship to texts' positions plus prior knowledge. When the gaze encoding is ablated, corresponding positional data and modules in Fig. 3 are removed. Similarly, knowledge-oriented inputs and embeddings will be removed when we do knowledge embedding ablation. As these ablations directly remove corresponding components in our model, each would reduce our model's parameter size. To further demonstrate our model succeeds in unknown word detection due to leveraging pre-trained RoBERTa weights rather than simply scaling the model's size up, we add an extra ablation ("w/o pretrained RoBERTa") where the RoBERTa model parameters are added to the model but randomly initialized.

4.1.4 N-Gram Baseline. Due to the overlap of unknown words between the training set and test set, the model might exhibit a shortcut by memorizing the unknown words in the training set. To examine the impact of this shortcut on model performance, we implemented an n-gram-based baseline to simulate the scenario where the model "memorizes" the unknown words. During prediction, if the current word combined with its preceding $n - 1$ words forms an n-gram appearing in the training set while also with the current word being labeled as unknown, the current word is classified as an unknown word at inference time. The larger the n, the more contextual information the n-gram baseline utilized. Since the degree of overlap in unknown words between the training and test sets varies across the main, cross-user, and cross-document settings, we applied the n-gram baseline ($n = 1, 2, 3$) to each of these settings individually (Table 4).

4.2 Model Performance

The main results of our experiment are shown in Table 2, the F1-score, precision and recall of our model that was trained and tested on data collected by eye tracker are 71.1%, 63.3% and 79.0%. Our model significantly outperforms the heuristic baseline (F1-score 22.9%), logistic regression (F1-score 23.4%) and the previous work based on SVM [16] (F1-score 29.9%). The performance of SVM is much worse than that presented in their paper. This could be caused by the smaller line space, which made it harder to assign the gaze to the word correctly. Besides, the document we used was more than two times longer than theirs, which led to a longer reading time (2.88 minutes per document and 8-14 minutes per reading session) and more posture changes of participants. This could worsen the tracking accuracy of eye tracker. Compared to their method, our method relies less on the gaze modality and PLMs also provide rich linguistic characteristics, thus improving the performance.

We also tested our model on the relatively inaccurate gaze data collected using a webcam (Fig. 6). To quantitatively compare the quality of gaze data collected by the eye tracker and webcam, we calculated the median absolute error (MAE) between these two types of gaze data. Since the sampling rate of the webcam is lower than that of the eye tracker, we applied cubic spline interpolation to the gaze coordination collected by webcam. The MAE in the x-direction is 115.99 pixels ($SD = 46.32$). In the y-direction, the MAE is 67.25 pixels ($SD = 32.28$). Considering that the height of the text is 20.63 pixels per line, the data from the webcam is

Table 3: Ablation study with eye tracker collected gaze data.

Method	Accuracy (%)	F1 (%)	Precision (%)	Recall (%)
Ours (main model)	97.6	71.1	63.3	79.0
w/o textual encoding	96.6	16.2	8.1	91.4
w/o gaze encoding	97.5	68.5	63.5	74.4
w/o knowledge embedding	97.6	69.2	68.4	70.0
w/o pretrained RoBERTa	97.7	67.6	61.9	74.6

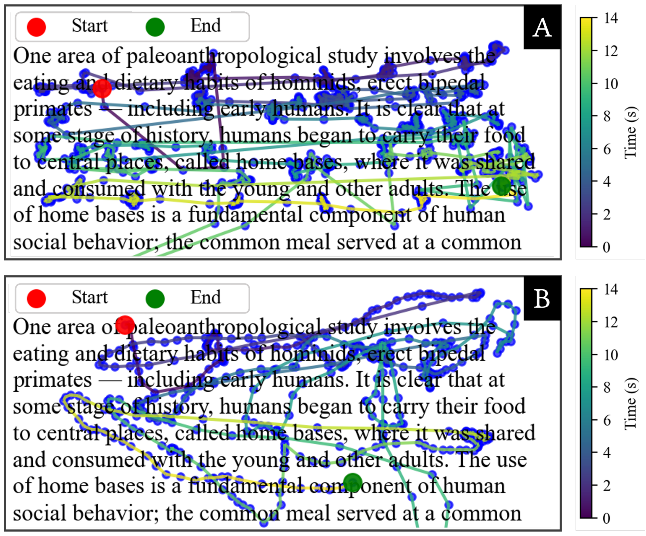


Figure 6: (A) Gaze data collected by a Tobii eye tracker. (B) Gaze data collected by a webcam.

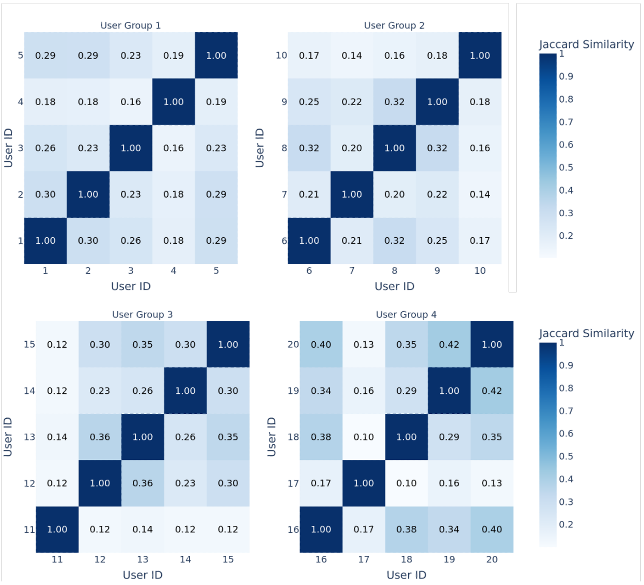


Figure 7: Jaccard similarity of unknown words between users.

noisier compared to the data obtained from the eye tracker. The model trained and tested on the relatively inaccurate gaze data collected using a Webcam (Fig. 6) also achieves a high F1-score (65.1%) compared to baselines, even though slightly worse compared to our method trained on eye tracker collected data. This is another evidence that our method has a higher tolerance for the noise of gaze data. This result also opens the potential for more accessible solutions than previously possible.

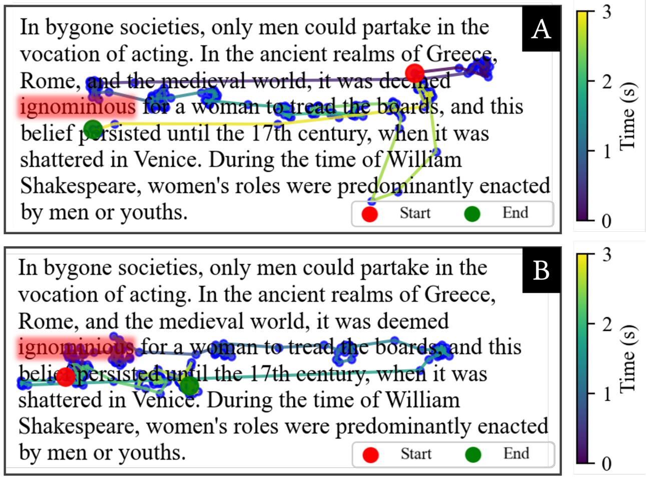
4.3 Contribution of Gaze

Firstly, we analyzed the differences in unknown words among users to demonstrate that the user-dependent gaze feature is necessary for correctly identifying unknown words for different users. We computed the Jaccard similarity matrix for all users. Users were divided into four groups because different groups of users read different documents. The average cross-user Jaccard similarity score is 0.24 and 77.5% of user-paired scores are below 0.3. The low Jaccard similarity score indicates that different users have different unknown words even reading the same document. Only relying on textual information cannot identify differences between users, so it is necessary to use the user-dependent gaze features.

Then, we removed the gaze encoding in the ablation study to determine the extent of the contribution of gaze to unknown word detection. After removing the gaze, the recall drops from 79.0% to 74.4% while the precision remains almost the same. This phenomenon implies that gaze is helpful for identifying an individual's unknown words based on the user's unique behaviors and patterns.

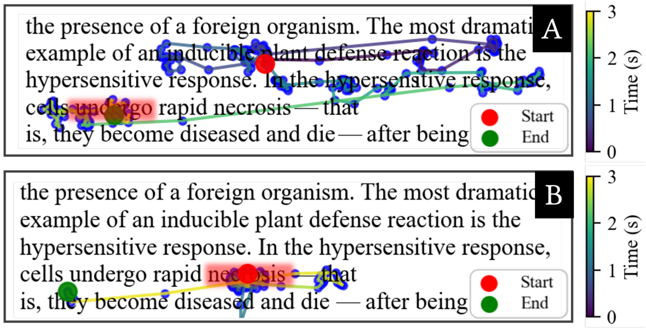
We also analyzed some cases to verify that gaze can help reflect differences between users and correctly predict unknown words in real time. We plotted the clip of gaze trajectory within 3 seconds (1-second gaze to locate the region of interest and additional 2-second gaze for features) and the corresponding text in the region of interest. In Case 1 shown in Fig. 8, the word "ignominious" is an unknown word for user B but is not an unknown word for user A. There is a noticeable dwell on "ignominious" in B's case which can facilitate the model to detect it as an unknown word correctly. The model captured this information because it predicted correctly in both cases.

Another case is shown in Fig. 9 in which the gaze in A and B are from the same user but in two consecutive seconds. The model correctly detects 'undergo' in the first second (A) and 'necrosis' in the second second (B) when the user's regions of interest in two consecutive seconds both contain 'undergo' and 'necrosis'. This result also indicates that our model learned gaze features and gaze plays an important role in real-time detection.


In bygone societies, only men could partake in the vocation of acting. In the ancient realms of Greece, Rome, and the medieval world, it was deemed ignominious for a woman to tread the boards, and this belief persisted until the 17th century, when it was shattered in Venice. During the time of William Shakespeare, women's roles were predominantly enacted by men or youths.

In bygone societies, only men could partake in the vocation of acting. In the ancient realms of Greece, Rome, and the medieval world, it was deemed ignominious for a woman to tread the boards, and this belief persisted until the 17th century, when it was shattered in Venice. During the time of William Shakespeare, women's roles were predominantly enacted by men or youths.

Figure 8: Case 1: (A) The word "ignominious" is not an unknown word for user A. (B) The word "ignominious" is an unknown word for user B.


the presence of a foreign organism. The most dramati example of an inducible plant defense reaction is the hypersensitive response. In the hypersensitive response, cells undergo rapid necrosis — that is, they become diseased and die — after being

the presence of a foreign organism. The most dramatic example of an inducible plant defense reaction is the hypersensitive response. In the hypersensitive response, cells undergo rapid necrosis — that is, they become diseased and die — after being

Figure 9: Case 2: (A) The word "undergo" was detected in the first second. (B) The word "necrosis" was detected in the second second.

4.4 Contribution of PLM

To identify the contribution of the pre-trained language model, we separately remove the "pre-trained" weights ("w/o pre-trained RoBERTa" in Fig. 3) by randomly initializing RoBERTa weights and remove the whole language model ("w/o textual encoding" in Fig. 3) by removing the RoBERTa weights. When initializing the RoBERTa randomly ("w/o pre-trained RoBERTa"), the F1-score decreases from 71.1% to 67.6% which indicates that the linguistic characteristics of words learned by the PLM help EyeLingo to detect unknown words more accurately. After entirely removing the RoBERTa parameters ("w/o textual encoding"), the F1-score significantly dropped to 22.0%, proving the effectiveness of using PLMs for better capturing the contextual information of the documents in the task of unknown word detection. The reason why the F1-score of "w/o pre-trained RoBERTa" is relatively high compared to "w/o textual encoding" is that the number of parameters remains constant in this case though the initialization is random, allowing RoBERTa to learn linguistic features incrementally during training.

Table 4: N-gram baseline (n=1, 2, 3) for main, cross-user and cross-document settings.

Method	F1 (%)	Precision (%)	Recall (%)
Main 1-gram	23.8	14.0	79.9
Main 2-gram	51.9	38.5	79.4
Main 3-gram	30.8	29.0	32.9
Cross-user 1-gram	19.3	11.4	62.5
Cross-user 2-gram	39.9	31.3	55.7
Cross-user 3-gram	19.1	16.9	22.1
Cross-doc 1-gram	7.9	4.9	21.0
Cross-doc 2-gram	5.9	9.0	4.5
Cross-doc 3-gram	3.5	4.8	2.8

Due to the overlap of unknown words between the training and test sets, we implemented an n-gram baseline as another naive baseline to simulate the scenario where the model 'memorizes' the unknown words to analyze the effect of this shortcut. As shown in Table 4, the results are best when $n = 2$, with an F1 score of 51.9%. This result is lower than our model (71.1%), indicating that merely memorizing words is insufficient. In contrast, our model leverages the PLM to provide a more comprehensive understanding of the linguistic information from context. Moreover, another interesting phenomenon is that the F1 score of the 2-gram baseline (51.9%) is higher than the gaze-based SVM baseline (29.9%). This suggests that textual information contributes more than gaze data when the line space is so small that the gaze is not accurate enough to locate individual words.

We also conducted the ablation study to the cross-document setting when the overlap of unknown words between the training and test sets is very small (Jaccard Similarity is 0.006). As shown in Table 6, removing gaze leads to a drop in performance, but it is less significant compared to the drop caused by the absence of pre-training.

Synthesizing the insights from these three experiments, we conclude that gaze is indispensable for locating regions of interest and providing real-time detection, and it also improves the model performance. However, the linguistic information introduced by the PLM plays the primary role in enhancing the model performance.

4.5 Cross-User and Cross-Document Generalizability

We evaluate our method with the cross-user generalizability with the leave-one-user-out setting, where the data collected from 18 randomly-selected users are set as the training dataset, while the rest of the two user's data are used as the dev and test sets. We find that in cross-user settings, the F1 score of our method drops to 59.6%, with the accuracy, precision, and recall decreasing to 97.3%, 52.8%, and 68.4%. Compared to the main setting, the performance decline of the model in the cross-user setting could be attributed to the loss of personalized information due to the absence of new user's gaze data. Another possible reason is the lack of labels for unknown words from new users, which affects the model's ability to assess word difficulty accurately for them. This suggests that it is necessary to collect some data from new users for fine-tuning

Table 5: Cross-user and Cross-document generalizability.

Method	Accuracy (%)	F1 (%)	Precision (%)	Recall (%)
Main	97.6	71.1	63.3	79.0
Cross-user	97.3	59.6	52.8	68.4
Cross-doc	96.8	51.1	43.3	62.8

before deployment to better adapt the model to different users. Additionally, relying heavily on the PLM might hinder the model's ability to achieve personalization.

To demonstrate that EyeLingo can indeed detect users' unknown words instead of simply memorizing difficult words in the documents, we also evaluate our model with the leave-one-document-out setting. In this case, we equally split the 120 reading documents into groups of 10, and in each training run, we trained our model on 110 documents while testing the model on the remaining 10 documents. The average Jaccard similarity score between the training set and test set is 0.006, demonstrating that the unknown words in the test set are very different from those in the training set. Our model's F1 score drops to 51.1%, with the precision, recall, and accuracy shifts to 43.3%, 62.8%, and 96.8%, respectively. The result indicates that generalizing the capabilities of unknown word detection across documents is challenging because of the lack of text context. Since the difference in context between the training and test set in the cross-document setting is larger than that in the cross-user setting, the performance in the cross-document setting is worse than cross-user setting. However, our model still achieves non-trivial improvements compared to the random classifier (F1-score 11.1%) and n-gram baseline (F1-score 7.9%), indicating that EyeLingo learns essential user gaze patterns and linguistic knowledge through training beyond naively memorizing difficult words.

To further investigate how gaze and PLM impact model performance, we conducted an ablation study in this challenging cross-document setting. After removing gaze and pre-training, the F1-score decreased to 49.9% and 37.4%, respectively. This result indicates that gaze contributes to the model's performance, but its impact is smaller compared to the contribution of textual information. Combined with the fact that the n-gram baseline's F1-score is only 7.9%, it can be concluded that the contribution of textual information lies not in merely memorizing words but in enhancing the model's understanding of context.

4.6 Latency and Memory Consumption

To demonstrate the capabilities of our method being used in real-time applications, we evaluate our method's latency with both CPU and GPU usage during inference. We set the batch size as 1 during inference latency testing. In GPU usage testing, we test the model with an RTX 4090 graphic card, where the average inference latency of the model is 0.013 seconds. Meanwhile, in CPU usage testing, the model's latency is 0.036 seconds. Overall, these results indicate that our method can support real-time applications with latency within 1 second. Moreover, under both settings, the maximum memory consumption of our model is 488.41MB, further proving that our method can be easily adapted to different on-device settings, widening our method's practicality for various downstream application supports.

5 User Evaluation

The user experience using the application built on EyeLingo depends on the real-time performance of EyeLingo. To evaluate the performance of our method in a real-time scenario, we built a reading assistance prototype that provided the translation and explanation of words while the user was reading. Then, we calculated the metrics such as F1-score and reading time of EyeLingo and also analyzed the subject feedback from users.

5.1 User Study Design

The questions we want to know the answers are

(1) What is the real-time performance (F1-score, correctly triggered rate, and false alarm rate) of our unknown word detection method (Section 5.3.1)?
(2) Can our method make reading more fluent (Section 5.3.2)?
(3) How is user experience when comparing our system to the commonly used method (Section 5.3.3)?

To answer these questions, we implemented three methods for real-time unknown word detection and compared them regarding objective metrics and subject scale. The first method (EyeLingo) uses gaze to locate the region of interest and applies our model to detect unknown words. When a new word is detected, its translation and explanation will be automatically displayed in the sidebar, and users do not need to take any action. The second method (Click) imitates the typical process of looking up words by clicking. When a user selects or double-clicks a word, a "meaning" option pops up for translation and explanation of the word. The third method (Ideal) simulates an ideal scenario where the model is highly accurate. Before users start reading, they are required to select unknown words from a list of candidate words. As users start reading, the words in the region of interest are compared with the selected unknown words. The system then displays the words in the unknown word list. To control variables other than prediction accuracy, all data processing and model inference procedures for Ideal are the same as those of EyeLingo. This approach aims to eliminate any degradation in user experience caused by model inaccuracies and to solely evaluate the potential of the unknown word detection method using gaze and text.

For the objective metrics, we calculated the accuracy, F1-score, correctly triggered rate, and false alarm rate for EyeLingo. We also recorded the reading time of users under three methods.

For user experience, we conducted the evaluation for the three reading assistance based on each method across the following five aspects using a 5-point Likert scale:

- Preference: To which degree do you prefer the application? 1 for “I prefer this one least”. 5 for “I prefer this one most”.
- Willingness to use: Would you like to use it in the future? 1 for “I won't use this application in the future”. 5 for “I would definitely use the application if it were available.”
- Usefulness: How helpful do you think it is for your reading and vocabulary learning? 1 for “Not helpful at all”. 5 for “Very helpful”.

Table 6: Ablation study on leave-one-document-out setting.

Method	Accuracy (%)	F1 (%)	Precision (%)	Recall (%)
Cross-document	96.8	51.1	43.3	62.8
w/o textual encoding	96.7	18.9	11.4	65.0
w/o gaze encoding	96.8	49.9	42.8	60.1
w/o knowledge embedding	96.7	34.0	26.6	49.6
w/o pretrained RoBERTa	96.7	34.7	26.6	51.0

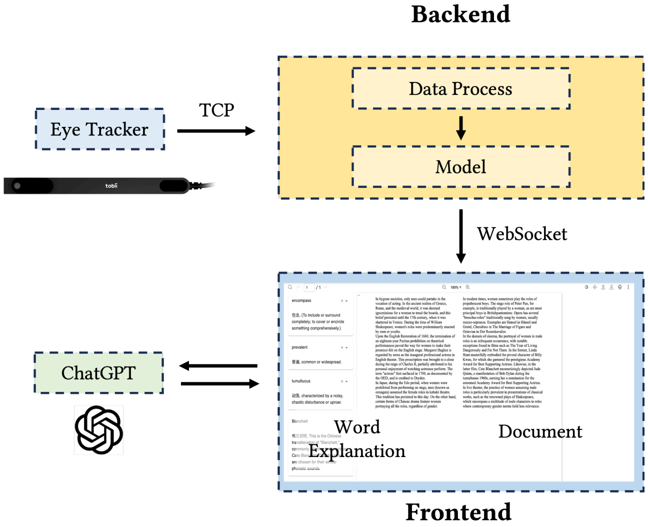
- Perceived reading fluency: How do you feel about your reading fluency when using this feature? 1 for “My reading is still slow and unsmooth”. 5 for “It speeds up my reading a lot and makes my reading very fluent”.
- Perceived latency: What do you think of the latency from when you need help to when the system pops up an explanation? 1 for “The latency is very small and does not affect usage”. 5 for “The latency is very large and makes the system not usable at all”.

Then we compared the rate for three methods using Wilcoxon signed-rank test at .05 significance level.

5.2 Setup and Procedure

5.2.1 Implementation of a Reading Assistance Proof-of-Concept. The proof-of-concept is a web-based application as shown in Fig. 10. The back-end was built using Python and the front-end was built using TypeScript and React. The React-PDF-viewer[7] was used to show the pdf. For the EyeLingo and Ideal, the gaze was captured by Tobii Nano which was the same model as the one used in data collection for model training. Next, the data was transmitted to the Python back-end via TCP and input into the unknown word detection model for inference along with text data. The model we used here is the main model trained on all the data and without any ablation. The same data processing step as the training was applied to gaze and text data. Then, unknown words predicted by the model were sent to the front-end through a web socket. The OpenAI API was called in the front-end to get translations and contextual explanations of words. In the end, the translations and explanations of words were displayed on the web page’s sidebar. For the Click, the unknown words were detected by a click listener and then called the OpenAI API to get the translations and explanations. The study was conducted on a MacBook Pro (CPU: Apple M1 Pro, RAM: 16G, screen: 14 inches).

5.2.2 Participant and Material. We recruited 10 English as a second language learners aged between 22-28 ($M = 23.80, SD = 1.87$), including 4 males and 6 females. Participants in this evaluation are different from participants for data collection. Eight of them wore glasses and two of them didn’t wear glasses. Their Vocabulary Levels Test (VLT) score on the 5000-word frequency level group ranged from 10 to 29 ($M = 18.80, SD = 5.07$).



1 / 1
150%

encompass

包含, {To include or surround completely; to cover or encircle something comprehensively.}

prevalent

普遍, common or widespread.

tumultuous

动荡, characterized by a noisy, chaotic disturbance or uproar.

Blanchett

布兰切特, This is the Chinese transliteration of "Blanchett," commonly [illegible] Cate Blanc[illegible] are chosen for their similar phonetic sounds.

In bygone societies, only men could partake in the vocation of acting. In the ancient realms of Greece, Rome, and the medieval world, it was deemed ignominious for a woman to tread the boards, and this belief persisted until the 17th century, when it was shattered in Venice. During the time of William Shakespeare, women's roles were predominantly enacted by men or youths.

Upon the English Restoration of 1660, the termination of an eighteen-year Puritan prohibition on theatrical performances paved the way for women to make their presence felt on the English stage. Margaret Hughes is regarded by some as the inaugural professional actress in English theater. This proscription was brought to a close during the reign of Charles II, partially attributed to his personal enjoyment of watching actresses perform. The term "actress" first surfaced in 1700, as documented by the OED, and is credited to Dryden.

In Japan, during the Edo period, when women were prohibited from performing on stage, men (known as onnagata) assumed the female roles in kabuki theatre. This tradition has persisted to this day. On the other hand, certain forms of Chinese drama feature women portraying all the roles, regardless of gender.

In modern times, women sometimes play the roles of prepubescent boys. The stage role of Peter Pan, for example, is traditionally played by a woman, as are most principal boys in Britishpantomime. Opera has several "breeches roles" traditionally sung by women, usually mezzo-sopranos. Examples are Hansel in Hänsel und Gretel, Cherubino in The Marriage of Figaro and Octavian in Der Rosenkavalier.

In the domain of cinema, the portrayal of women in male roles is an infrequent occurrence, with notable exceptions found in films such as The Year of Living Dangerously and I'm Not There. In the former, Linda Hunt masterfully embodied the pivotal character of Billy Kwan, for which she garnered the prestigious Academy Award for Best Supporting Actress. Likewise, in the latter film, Cate Blanchett mesmerizingly depicted Jude Quinn, a manifestation of Bob Dylan during the tumultuous 1960s, earning her a nomination for the esteemed Academy Award for Best Supporting Actress.

In live theater, the practice of women assuming male roles is particularly prevalent in presentations of classical works, such as the renowned plays of Shakespeare, which encompass a multitude of male characters in roles where contemporary gender norms hold less relevance.

Figure 10: The implementation of reading assistance prototype based on our unknown word detection method.

The documents we used are from TOEFL reading material with similar lengths (390 words/24 lines, 353 words/23 lines, 371 words/22 lines). Another important reason for selecting these three documents is that, their total numbers of unknown words (49 words, 49 words, 52 words) marked in the previous data collection are close. During the user study, the text was displayed in two columns, consistent with the settings used for data collection. We employed Times New Roman font at size 10 with single line spacing.

5.2.3 Procedure. Firstly, the experimenter introduced the background of the study and randomly assigned three articles to three tasks (EyeLingo, Click, and Ideal). Afterward, the eye tracker was calibrated and the participants were required to try the EyeLingo and Click to get familiar with them. After the calibration, participants were informed to maintain the same posture as much as possible and read with their original reading habits.

Participants used EyeLingo when reading the first document. The application automatically saved the unknown words detected by the model, the user’s reading time, and gaze data. Participants marked unknown words they encountered for ground truth after the reading. Participants used Click and got word explanation by clicking the word when reading the second document. The reading time was recorded. For the Ideal, participants first deleted words they knew from the unknown word candidate list before the reading. The gaze function was turned to locate the region of interest

[7] https://react-pdf-viewer.dev/

and the explanations for words from the unknown word list were displayed when the participant encountered those words. Reading time and gaze data were recorded. After reading three documents, the participants were asked to rate EyeLingo, Click, and Ideal individually in the first aspects mentioned in Section. 5.1.

5.3 Analysis and Results

5.3.1 Real-Time Performance. We calculated the F1-score, precision, and recall for each participant by comparing the unknown words marked by the participants and detected by the model in EyeLingo setting. The averaged F1-score, precision, and recall are respectively 56.54%, 51.93%, and 67.39%, which is close to the cross-user result in Section 4.5 (59.6%, 52.8%, and 68.4%). This result shows that our model performs the same when running offline and in real-time, which proves that our model can work in real-time. The average correctly triggered rate is equal to recall which is 67.39% and the average false alert is 2.89%. The false alert rate is similar to the previous work conducted using the more accurate head-mounted eye tracker [24]. The low correctly triggered rate could be caused by the inaccuracy of our model and the inevitable change of posture when the user was reading.

5.3.2 Reading Fluency. We evaluated the influence of our word detection method on improving users' reading fluency from two perspectives which are reading time and user perceived fluency. The reading time of EyeLingo ($p = 0.006$) and Ideal ($p = 0.048$) is significantly shorter than the reading time of Click. There is no significant difference between the reading time of EyeLingo and Ideal ($p = 0.193$).

For the perceived reading fluency, the rate of ideal is significantly higher than Click ($p = 0.030$), but there is no significant difference between EyeLingo and Click ($p = 0.0773$). Automatically detecting unknown words speeds up the reading by minimizing the number of operations. However, participants' feedback shows that the explanation of incorrect unknown words that popped up in EyeLingo distracted users and slowed down the reading when using EyeLingo. Ideal does not have this problem and will not interrupt reading due to mouse operation, so it makes the reading process more fluent.

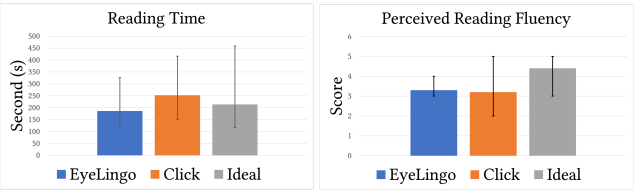


Figure 11: EyeLingo enhances the fluency of reading for users compared with getting word explanation by clicking the unknown word.

5.3.3 User Experience. For user preference, the average scores for Click and Ideal are the same as shown in Fig. 12. It indicates that detecting unknown words by gaze and PLMs has the potential to replace the conventional method of word lookup via mouse clicks. In terms of willingness to use, the score of Ideal (4.1) is higher than Click (3.5). Most participants stated they preferred real-time automatic word detection if the accuracy was high. For the usefulness, the score of Ideal (4.2) is slightly higher than Click (4.0). Participants indicated that our proposed method helps them read more fluently with less disruption. The perceived latency of Ideal (2.7) is lower than Click (3.1) demonstrating that the latency caused by the data processing and model inference is negligible because the Ideal also includes these components from EyeLingo. Moreover, the time saved by avoiding clicks can compensate for the delay introduced by GPT, thereby reducing overall latency.

The scores of EyeLingo are lower than Click, but Ideal can surpass traditional click methods in most aspects. Considering that the major difference between EyeLingo and Ideal is the accuracy of the unknown word detection model, EyeLingo could offer a more natural and efficient interaction mechanism by further improving the accuracy in the future.

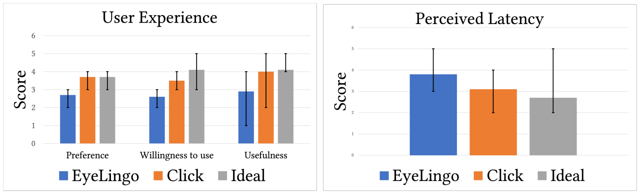


Figure 12: Subjective rating of user experience. Left: the ideal condition of our method improves the willingness to use and usefulness compared to the click method. Right: the latency that participants perceived of ideal condition is less than click.

6 Discussion

In this section, we first summarize the conclusion and share some key observations. Then, we reflect on the usability of our method and propose potential applications. In the end, we discuss the limitations and future work.

6.1 Effectiveness of EyeLingo

Firstly, based on the results from Section 4, we can draw the following conclusions:

- It is efficient to detect unknown words by combining linguistic characteristics provided by the pre-trained language model (PLM) and gaze trajectory.
- The prediction is mainly based on the linguistic features from the textual context captured by PLM.
- Gaze locates the region of interest in a timely manner, which is necessary for real-time applications. Gaze also helps improve the model performance, but its contribution is limited compared to PLM.

Additionally, it is interesting that while we typically assume that the gaze modality should contribute significantly to the task of unknown word detection, the experimental results show that the contribution of gaze to the model's improvement is small with the existence of PLM. Based on the previous analysis of line spacing and eye tracker accuracy, a possible reason for this is that under normal reading conditions (single-line spacing, line height 3-5 mm), the eye tracker's accuracy is insufficient to precisely detect which line the gaze belongs to, thus failing to accurately locate the gaze on the words. Furthermore, changes in user posture during long reading sessions further reduce the accuracy of the eye tracker. In our system, PLM compensates for this issue by providing linguistic information based on the text.

From another perspective, the low contribution of gaze is not necessarily a disadvantage. Our method's reduced reliance on gaze makes it more tolerant of noise. The model's good performance on data collected by webcams further supports this conclusion. The reduced dependency on gaze data allows our model to be applied on more affordable and accessible devices, such as webcams.

6.2 Usability of EyeLingo

The results from the user evaluation (Section 5) show that our reading assistance prototype helps users read more fluently and they are more willing to use it compared to traditional click-to-translate methods. In addition to providing real-time translation and explanations during reading, our system can also benefit ESL for long-term learning. For example, based on the unknown word detected by our system, we can generate a vocabulary list for memorizing and offer memory curve tracking. Furthermore, these unknown words can also be used to generate personalized summaries and notes.

The potential issue of generalizability across users, texts and devices can be addressed through fine-tuning and reinforcement learning methods. During the initial phases of usage, the system collects both gaze and text data for fine-tuning and lets users provide feedback on the model's predictions. This allows the model to continuously learn the user's unique gaze patterns and infer their vocabulary proficiency and domain expertise from textual content, thereby improving prediction accuracy.

6.3 Limitation and Future Works

The quality of gaze data hinders the improvement model performance. The accuracy of the eye tracker is not enough for word-level detection. Common formatting, such as single-line spacing and 10-point font, results in a line height of approximately 3-5 mm when viewed using the PDF viewer with a sidebar on a 14-inch laptop. This requires an accuracy of about $0.3 - 0.6^\circ$ at a reading distance of 50-60 cm. However, most eye trackers have a gaze accuracy ranging from $0.2 - 1.1^\circ$ [5]. Combined with additional errors caused by head and upper body movements, this level of accuracy is insufficient for real-world reading scenarios. During data collection and evaluation, some participants reported that even after calibration, the error could span 1-3 lines. This makes it difficult to determine the specific word the user is focusing on based solely on gaze coordinates, explaining why gaze-based baselines performed poorly on our data.

The inaccuracy of the gaze data could also lead to the inaccuracy of data labeling. To mitigate the impact of mouse clicks on gaze behavior, we asked users to label unknown words during their second pass. However, this widely adopted labeling method inherently requires "guessing" which words correspond to a given gaze trajectory. Previous works mapped each gaze coordinate directly to a specific word to establish word-gaze pairs. This method is infeasible for text with normal line spacing, so we establish gaze-word pairs by defining a bounding box based on a segment of gaze to identify the corresponding words instead. While this approach improves robustness, it may also introduce mismatches between gaze and words and thus introduce noise to the dataset. To further improve model performance, more precise labeling methods are needed.

Additionally, reading time can be longer than several minutes in daily scenarios, so gaze drift can significantly affect data quality. In our experiments, we observed that it is difficult for participants to maintain a fixed posture after calibration, though we required them to do so. The posture shift further increases errors. Therefore, in practical applications, real-time calibration of gaze data based on user posture is crucial to ensure data quality. If the existing eye-tracking technology can combined with user posture detection [44], it is possible to reduce the impact of user posture on gaze data, thereby improving the quality of gaze data.

7 Conclusion

We propose a highly accurate and real-time unknown-word detecting method based on gaze trajectory and pre-trained language models (PLMs). The text embedding derived from the PLM and the knowledge grounding provides rich linguistic characteristics for candidate words, and the gaze data locates the region of interest and supplies behavior information to the model. The evaluation shows that it achieves an accuracy of 97.6% and F1-score of 71.1%. The latency is within 1 second. Our method also works on noisy gaze data acquired by webcam with an accuracy of 97.3% and an F1-score of 65.1%, which demonstrates the accessibility and applicability for daily use. The experimental results show that the PLM contributes most to the performance and gaze provides user-dependent features. The real-time evaluation shows that our method achieves the F1-score of 56.54% and makes reading more fluent. It also shows an improvement in the willingness to use and usefulness compared to the traditional click method. Due to the growing number of embedded eye tracking in commercial devices, we believe that lots of language learners can ultimately benefit from our technique.

Acknowledgments

This work is supported by the Natural Science Foundation of China under Grant No. 62132010, 62472244, 62102221, the Tsinghua University Initiative Scientific Research Program, and the Undergraduate Education Innovation Grants, Tsinghua University.

References

[1] Maria Barrett, Joachim Bingel, Nora Hollenstein, Marek Rei, and Anders Søgaard. 2018. Sequence Classification with Human Attention. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Brussels, Belgium, 302–312. https://doi.org/10.18653/v1/K18-1030

[2] Maria Barrett and Nora Hollenstein. 2020. Sequence labelling and sequence classification with gaze: Novel uses of eye-tracking data for Natural Language Processing. *Language and Linguistics Compass* 14, 11 (2020), 1–16.

[3] R. Bates and H. O. Istance. 2003. Why Are Eye Mice Unpopular? A Detailed Comparison of Head and Eye Controlled Assistive Technology Pointing Devices. *Univers. Access Inf. Soc.* 2, 3 (oct 2003), 280–290. https://doi.org/10.1007/s10209-003-0053-y

[4] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

[5] Carmen Bisogni, Michele Nappi, Genoveffa Tortora, and Alberto Del Bimbo. 2024. Gaze analysis: A survey on its applications. *Image and Vision Computing* 144 (2024), 104961. https://doi.org/10.1016/j.imavis.2024.104961

[6] Robert Bixler and Sidney D'Mello. [n. d.]. Automatic gaze-based user-independent detection of mind wandering during computerized reading. 26, 1 ([n. d.]), 33–68. https://doi.org/10.1007/s11257-015-9167-1

[7] Zhaokang Chen and Bertram E. Shi. 2019. Using Variable Dwell Time to Accelerate Gaze-Based Web Browsing with Two-Step Selection. *International Journal of Human–Computer Interaction* 35, 3 (2019), 240–255. https://doi.org/10.1080/10447318.2018.1452351 arXiv:https://doi.org/10.1080/10447318.2018.1452351

[8] Shiwei Cheng, Zhiqiang Sun, Lingyun Sun, Kirsten Yee, and Anind K. Dey. [n. d.]. Gaze-Based Annotations for Reading Comprehension. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul Republic of Korea, 2015-04-18). ACM, 1569–1572. https://doi.org/10.1145/2702123.2702271

[9] Kiyomi Chujo and Kathryn Oghigian. 2009. How many words do you need to know to understand TOEIC, TOEFL & EIKEN? An examination of text coverage and high frequency vocabulary. *Journal of Asia TEFL* 6, 2 (2009).

[10] Shuwen Deng, David R. Reich, Paul Prasse, Patrick Haller, Tobias Scheffer, and Lena A. Jäger. 2023. Eyettention: An Attention-based Dual-Sequence Model for Predicting Human Scanpaths during Reading. *Proc. ACM Hum.-Comput. Interact.* 7, ETRA, Article 162 (May 2023), 24 pages. https://doi.org/10.1145/3591131

[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)*. 4171–4186.

[12] Piaoyang Du, Wei Guo, and Shiwei Cheng. [n. d.]. Using eye-tracking for real-time translation: a new approach to improving reading experience. 6, 2 ([n. d.]), 150–164. https://doi.org/10.1007/s42486-024-00150-3

[13] Neeru Dubey, Simran Setia, Amit Arjun Verma, and S. R.S. Iyengar. 2020. WikiGaze: Gaze-Based Personalized Summarization of Wikipedia Reading Session. In *Proceedings of the 3rd Workshop on Human Factors in Hypertext* (Virtual Event, USA) *(HUMAN'20)*. Association for Computing Machinery, New York, NY, USA, Article 4, 9 pages. https://doi.org/10.1145/3406853.3432662

[14] Paola Dussias. 2010. Uses of Eye-Tracking Data in Second Language Sentence Processing Research. *Annual Review of Applied Linguistics* 30 (03 2010), 149 – 166. https://doi.org/10.1017/S026719051000005X

[15] Yo Ehara, Nobuyuki Shimizu, Takashi Ninomiya, and Hiroshi Nakagawa. 2010. Personalized Reading Support for Second-Language Web Documents by Collective Intelligence. In *Proceedings of the 15th International Conference on Intelligent User Interfaces* (Hong Kong, China) *(IUI '10)*. Association for Computing Machinery, New York, NY, USA, 51–60. https://doi.org/10.1145/1719970.1719978

[16] Utpal Garain, Onkar Pandit, Olivier Augereau, Ayano Okoso, and Koichi Kise. 2017. Identification of Reader Specific Difficult Words by Analyzing Eye Gaze and Document Content. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Vol. 01. 1346–1351. https://doi.org/10.1109/ICDAR.2017.221

[17] Wei Guo and Shiwei Cheng. 2019. An Approach to Reading Assistance with Eye Tracking Data and Text Features. In *Adjunct of the 2019 International Conference on Multimodal Interaction* (Suzhou, China) *(ICMI '19)*. Association for Computing Machinery, New York, NY, USA, Article 7, 7 pages. https://doi.org/10.1145/3351529.3360659

[18] Brett J. Hashimoto and Jesse Egbert. 2019. More Than Frequency? Exploring Predictors of Word Difficulty for Second Language Learners. *Language Learning* 69, 4 (2019), 839–872. https://doi.org/10.1111/lang.12353 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/lang.12353

[19] Riku Higashimura, Andrew Vargo, Motoi Iwata, and Koichi Kise. 2022. Helping Mobile Learners Know Unknown Words through Their Reading Behavior. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI EA '22)*. Association for Computing Machinery, New York, NY, USA, Article 249, 5 pages. https://doi.org/10.1145/3491101.3519620

[20] Rui Hiraoka, Hiroki Tanaka, Sakriani Sakti, Graham Neubig, and Satoshi Nakamura. 2016. Personalized Unknown Word Detection in Non-Native Language Reading Using Eye Gaze. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction* (Tokyo, Japan) *(ICMI '16)*. Association for Computing Machinery, New York, NY, USA, 66–70. https://doi.org/10.1145/2993148.2993167

[21] Nora Hollenstein, Marius Troendle, Ce Zhang, and Nicolas Langer. 2020. ZuCo 2.0: A Dataset of Physiological Recordings During Natural Reading and Annotation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis (Eds.). European Language Resources Association, Marseille, France, 138–146. https://aclanthology.org/2020.lrec-1.18

[22] Nora Hollenstein and Ce Zhang. 2019. Entity Recognition at First Sight: Improving NER with Eye Movement Information. In *North American Chapter of the Association for Computational Linguistics*.

[23] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength natural language processing in python. (2020).

[24] Aulikki Hyrskykari. 2006. Eyes in attentive interfaces : experiences from creating iDict, a gaze-aware reading aid /. (01 2006).

[25] Marcel A Just and Patricia A Carpenter. 1980. A theory of reading: from eye fixations to comprehension. *Psychological review* 87, 4 (1980), 329.

[26] Jiajia Li, Grace Ngai, Hong Va Leong, and Stephen Chan. [n. d.]. Multimodal human attention detection for reading. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing* (Pisa Italy, 2016-04-04). ACM, 187–192. https://doi.org/10.1145/2851613.2851681

[27] Change Liu, Chun Yu, Xiangyang Wang, Jianxiao Jiang, Tiaoao Yang, Bingda Tang, Yingtian Shi, Chen Liang, and Yuanchun Shi. 2024. CalibRead: Unobtrusive Eye Tracking Calibration from Natural Reading Behavior. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8, 4, Article 154 (Nov. 2024), 30 pages. https://doi.org/10.1145/3699737

[28] Jiahui Liu, Jiannan Chi, Huijie Yang, and Xucheng Yin. 2022. In the eye of the beholder: A survey of gaze tracking techniques. *Pattern Recognition* 132 (2022), 108944. https://doi.org/10.1016/j.patcog.2022.108944

[29] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[30] Yunfei Long, Rong Xiang, Qin Lu, Chu-Ren Huang, and Minglei Li. 2021. Improving Attention Model Based on Cognition Grounded Data for Sentiment Analysis. *IEEE Transactions on Affective Computing* 12, 4 (2021), 900–912. https://doi.org/10.1109/TAFFC.2019.2903056

[31] Ahmad Azman Mokhtar and Rafizah Mohd Rawian. 2012. Guessing word meaning from context has its limit: Why. *International Journal of Linguistics* 4, 2 (2012), 288–305.

[32] Paul Nation. 1990. Teaching and learning vocabulary. In *Handbook of Practical Second Language Teaching and Learning*. Routledge, 397–408.

[33] Ayano Okoso, Takumi Toyama, Kai Kunze, Joachim Folz, Marcus Liwicki, and Koichi Kise. [n. d.]. Towards Extraction of Subjective Reading Incomprehension: Analysis of Eye Gaze Features. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems* (Seoul Republic of Korea, 2015-04-18). ACM, 1325–1330. https://doi.org/10.1145/2702613.2732896

[34] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.

[35] Keith Rayner. 1995. Eye Movements and Cognitive Processes in Reading, Visual Search, and Scene Perception. In *Eye Movement Research*, John M. Findlay, Robin Walker, and Robert W. Kentridge (Eds.). Studies in Visual Information Processing, Vol. 6. North-Holland, 3–22. https://doi.org/10.1016/S0926-907X(05)80003-0

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

[37] Pat Rigg. 1991. Whole language in TESOL. *Tesol Quarterly* 25, 3 (1991), 521–542.

[38] Charles Sanches, Olivier Augereau, and Koichi Kise. 2016. Vertical error correction of eye trackers in nonrestrictive reading condition. *IPSJ Transactions on Computer Vision and Applications* 8 (12 2016). https://doi.org/10.1186/s41074-016-0008-x

[39] Charles Lima Sanches, Olivier Augereau, and Koichi Kise. [n. d.]. Using the Eye Gaze to Predict Document Reading Subjective Understanding. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)* (Kyoto, 2017-11). IEEE, 28–31. https://doi.org/10.1109/ICDAR.2017.377

[40] Norbert Schmitt, Diane Schmitt, and Caroline Clapham. 2001. Developing and exploring the behaviour of two new versions of the Vocabulary Levels Test. *Language testing* 18, 1 (2001), 55–88.

[41] John L. Sibert, Mehmet Gokturk, and Robert A. Lavine. [n. d.]. The reading assistant: eye gaze triggered auditory prompting for reading remediation. In *Proceedings of the 13th annual ACM symposium on User interface software and technology - UIST '00* (San Diego, California, United States, 2000). ACM Press, 101–107. https://doi.org/10.1145/354401.354418

[42] Ekta Sood, Simon Tannert, Philipp Mueller, and Andreas Bulling. 2020. Improving Natural Language Processing Tasks with Human Gaze-Guided Neural Attention. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 6327–6341. https://proceedings.neurips.cc/paper_files/paper/2020/file/460191c72f67e90150a093b4585e7eb4-Paper.pdf

[43] 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 International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6000–6010.

[44] Yuntao Wang, Jiexin Ding, Ishan Chatterjee, Farshid Salemi Parizi, Yuzhou Zhuang, Yukang Yan, Shwetak Patel, and Yuanchun Shi. 2022. FaceOri: Tracking Head Position and Orientation Using Ultrasonic Ranging on Earphones. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 290, 12 pages. https://doi.org/10.1145/3491102.3517698

[45] Akito Yamaya, Goran Topić, Pascual Martínez-Gómez, and Akiko Aizawa. 2015. Dynamic-Programming–Based Method for Fixation-to-Word Mapping. In *Intelligent Decision Technologies*, Rui Neves-Silva, Lakhmi C. Jain, and Robert J. Howlett

(Eds.). Springer International Publishing, Cham, 649–659.

[46] Francesca Zermiani, Prajit Dhar, Ekta Sood, Fabian Kögel, Andreas Bulling, and Maria Wirzberger. 2024. InteRead: An Eye Tracking Dataset of Interrupted Reading. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (Eds.). ELRA and ICCL, Torino, Italia, 9154–9169. https://aclanthology.org/2024.lrec-main.802