Facilitating Gaze-based Unknown Word Detection Using Pre-trained Language Models (PLMs)

ANONYMOUS AUTHOR(S)

As non-native speakers learn a new language, encountering words outside one's vocabulary can impact their reading fluency. Automatically detecting these unknown words can help drive interactive methods to support reading. Previous methods of unknown word detection methods rely on gaze features obtained through eye trackers, with their detection accuracy greatly affected by the accuracy of eye tracking devices. In this work, we present a real-time and high-accuracy unknown word detection method by combining information about the text with gaze data. We utilize pre-trained language models and knowledge grounding to analyze text for unknown word probabilities. We then combine this text data with gaze information through a transformer-based model. The accuracy of our unknown word detection method is 97.6%, and the F1-score is 71.1%. The latency is within 1 second. To demonstrate the robustness of our method, we applied our method to another dataset collected with a relatively inaccurate webcam-based eye-tracking system. Our model can achieve the accuracy of 97.3% and the F1-score of 65.1% opening the potential for more accessible solutions than previously possible. Finally, we explore the applications space that can be enabled by our method.

CCS Concepts: • **Do Not Use This Code → Generate the Correct Terms for Your Paper**; *Generate the Correct Terms for Your Paper*; Generate the Correct Terms for Your Paper; Generate the Correct Terms for Your Paper.

Additional Key Words and Phrases: Unknown word detection, gaze, pre-trained language model.

ACM Reference Format:

Anonymous Author(s). 2018. Facilitating Gaze-based Unknown Word Detection Using Pre-trained Language Models (PLMs). *J. ACM* 37, 4, Article 111 (August 2018), 14 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Unknown words can greatly reduce reading fluency and worsen the reading experience of non-native speakers [17, 22]. By automatically detecting these unknown words, we can assist users in reading and language comprehension, and make their lives more convenient in non-native language environments. Previous methods mainly leverage gaze features such as fixation duration, number of fixations, and saccade length [24], since there is a correlation between gaze and word difficulty [13]. 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 constant eye motions, making it hard to precisely map a gaze point to the word in the text being read [1]. To reduce reliance on gaze information, other works seek to compensate or replace inaccurate eye-tracking data on commercial devices by incorporating text [8, 10, 12], click [7] and motion data [9].

With the development of Natural Language Processing (NLP) technology, we explore how to take advantage of the language model to make gaze-based unknown word detection accurate, easily accessible, and more applicable. We present a real-time unknown word detection method that leverages the text information processed by the language

Manuscript submitted to ACM

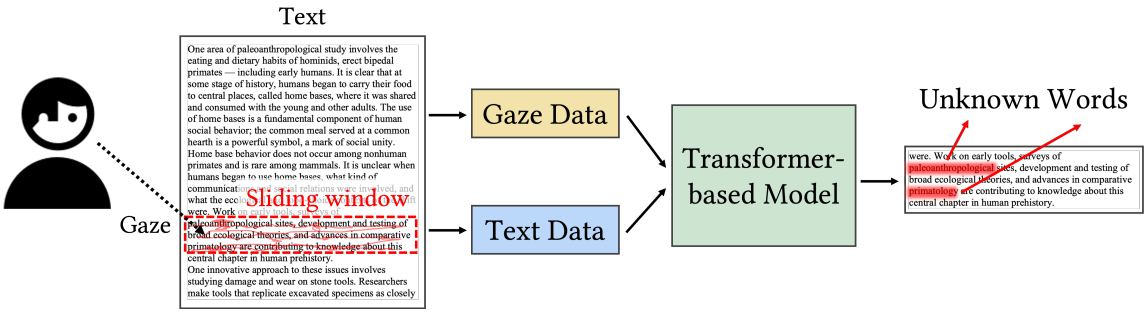

Fig. 1. Our method locates the content the user is reading in real-time through gaze, and inputs the gaze data and text data to the transform-based model to detect unknown words.

model to facilitate gaze-based detection. With the language model, the inaccuracy of the gaze-based method can be compensated by the probability derived from the language model based on the text. First, we verified the feasibility of our approach using professional eye trackers. Then, we applied our unknown word detection method to the webcam-based gaze data to show the robustness of our method even with commodity deployments.

As shown in Fig. 1, we tackle the problem of unknown word detection with two modalities of information during users' reading, i.e., the positional information of the document and the user's gaze, along with the contextual encoding of the document processed with a pre-trained language model. We first identify the sliding window of the user's gaze based on the readouts of the eye tracker or webcam. After extracting several key features from the user's gaze (position, word distance, fixation duration, etc.), we use a transformer-based encoder-decoder model with cross-attention modules to encode the positional information of the user's gaze within the document. In parallel, we also leverage a pre-trained language model to encode the textual information 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 a reading assistant in foreign language environments can be enabled.

Our contributions in this work are summarized as follows:

(1) We propose a real-time gaze-based unknown word detection method that integrates the text information processed by the pre-trained language model. It can achieve the accuracy of 97.6% and the F1-score of 71.1%. The latency is within 1 second.
(2) We demonstrate that our method has the potential to be easily accessed in everyday life through a webcam-based unknown word detection system. The accuracy of our model on the gaze data obtained by webcam is 97.3% and the F1-score is 65.1%.
(3) We illustrate that our method can be applied in a wide variety of scenarios through several typical applications.

2 RELATED WORK

In this section, we illustrate how gaze behavior is related to attention in reading and examine the reading behavior analysis enabled by tracking gaze. Then, we explain the limitation of the gaze-based unknown word detecting method and how the previous works improve their performance. After that, we review the cutting-edge unknown word detecting methods and analyze the gaps in detecting unknown words accurately.

2.1 Gaze in Reading

Reading as a cognitive process affects eye movement [13]. When people are reading, the the eyes follow the text through small amplitudes, ballistic motions called saccades. The pauses between two saccades are called a fixation [12]. As per the eye-mind hypothesis, the fixation on a word persists during its processing phase [13]. 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 [4, 18, 23], detecting mind wandering [3], identifying interest [6], and detecting attention [14] in reading.

Extended periods of fixations on the focal word suggest difficulties in word identification [12], forming the theoretical basis for detecting unknown words based via gaze. However, it is hard to achieve high accuracy only based on the fixation because of the inaccuracy of gaze-tracking hardware and algorithms, and the ambiguous relationship between gaze patterns and the cognitive processes of words. The highest accuracy of eye tracking is 0.3° (2.6 mm when the distance is 50cm) in optimal conditions, but the accuracy can be easily affected by the calibration performance and user posture [15]. Thus, how to accurately match the gaze point to the text is a problem. Additionally, other researchers pointed out that the processing of words can happen when they are not held in fixation [21].

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 [8, 10].

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 [12] detects the unknown word based on gaze duration and word frequency and sets a threshold to trigger a gloss or margin note on problematic words. It successfully detects 36.5% of problematic words. Later works extend iDict by replacing the threshold function with machine learning. Hiroka et al. [10] 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 is 55.6%) when adding linguistics features including word length and word rarity. Similarly, Garain et al. [8] also takes both gaze and linguistics features into consideration to achieve the best F1-score of 86% on a single user using an SVM. However, all these methods rely on a dedicated eye tracker to obtain accurate enough gaze data to extract the gaze features and match these features to the words. Furthermore, in their experimental design, the line spacing was set between 3.0 to 6.0, which makes it hard to apply their methods to real-world scenarios.

Apart from gaze, other reading behaviors such as mouse clicks and hand motions can be used to detect unknown words. Ehara et al. [7] 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. [9] is targeted 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 accurate gaze data via expensive eye trackers within laboratory environments. A real-time method

that is robust to relatively inaccurate gaze data from a webcam can promote the availability of unknown-word detection. We seek to build upon the rapid development of natural language processing (NLP) technology in recent years to yield such a solution. Pre-trained language models such as [5, 16] encompass extensive text information for downstream applications. We seek to harness their capabilities in a new architecture to increase the tolerance of inaccuracy in gaze tracking. We propose a transformer-based model that requires information from a sliding window around the gazed words.

3 UNKNOWN WORD DETECTION METHOD

In the section below, we first evaluate our method on the data collected using a dedicated eye tracker to substantiate its efficacy. Then we apply our method on the webcam-based data to demonstrate its robustness and availability.

3.1 Data Collection

3.1.1 Implementation. We built a system to collect gaze data through eye trackers and webcams at the same time. We used Tobii Pro Nano [1] eye tracker whose sampling rate is 60Hz and streamed its data to the computer through a Python script. For webcam data, we used a SeeSo [2], a remote eye tracking platform, by integrating it into our PDF reader. To get the text information including their contents and their 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 are Macbook Pro (CPU: Apple M1 Pro, RAM: 16GB, Storage: 512GB) and Huawai MateBook D14 2022 (CPU: i5-1155G7, RAM: 16GB, Storage: 512GB).

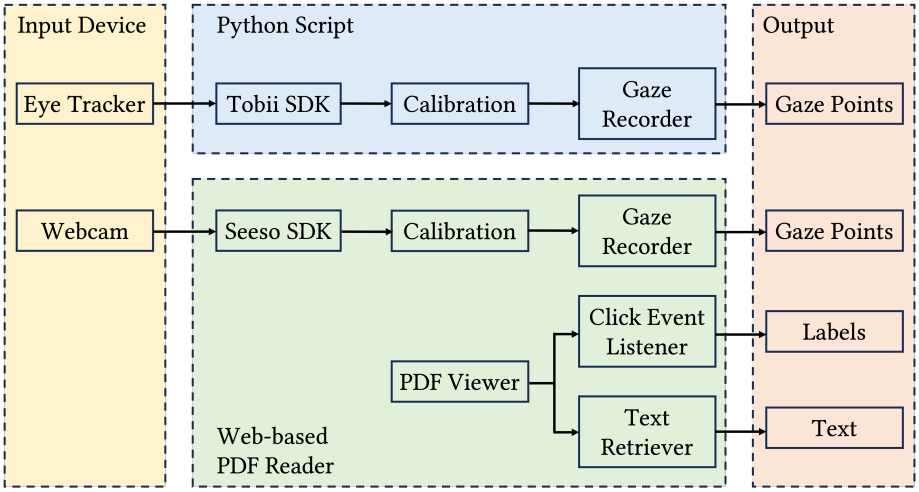

Fig. 2. The data collection platform includes a Python script to read eye tracker data and a web-based PDF reader to read webcam data and record word labeling.

[1] https://www.tobii.com/products/eye-trackers/screen-based/tobii-pro-nano
[2] https://seeso.io/
[3] https://github.com/zotero/pdf-reader

3.1.2 Participant and Material. We recruited a total of 20 undergraduate and graduate students (5 females, 15 males) whose second language is English. Their ages range from 21 to 26 years old (M = 22.85, SD = 1.65), and the number of years they have studied English formally ranges from 9 to 21 years (M = 18.70, SD = 3.27). Among them, 16 people wore glasses during the data collection and 4 did not. The reading materials contain 120 articles from 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.

3.1.3 Experiment Design and Procedure. Twenty participants were divided 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. We divided the reading into 3 sessions to ensure that the participants weren't fatigued by the lengthy reading. There was a calibration session before the data collection started for each participant each day to calibrate both the eye tracker and webcam eye tracker. Participants can choose to take a break or not after reading the fifth article according to their level of fatigue. If they chose to take a break, there would be re-calibration before the collection was restarted, considering that the person's sitting posture had a great impact on the accuracy of eye tracking.

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. 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.

3.2 Data Preprocessing

To support unknown word detection in real time, 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. 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.

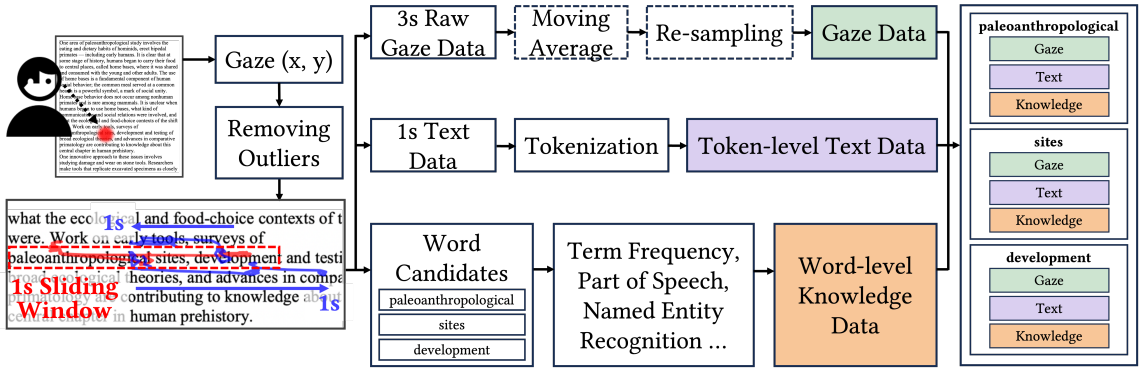

Fig. 3. A bounding box is derived from the gaze coordination within a 1-second sliding window. The gaze data, token-level text data, and word-level knowledge data are calculated for each candidate word in the bounding box.

As shown in the left part of Fig. 3, 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 (*average, std*). When the distance between a small subset of the data in a window and the other part of the data (*definition*) 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. 3), 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 into 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 [20] 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 frequency, part of speech, and named entity recognition to each candidate word. In the example shown in Fig. 3, 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.

3.3 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 architect to encode the positional gaze and text information into the vector space. We also leverage the RoBERTa [16] to integrate the 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. 4.

3.3.1 Positional Data Encoding. To accurately capture the correlation between the user's gaze and the document, we adapt the model architecture from T5 [19], 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 averaged gaze-token distance $d(g, w)$ and the approximated fixation time $t(g, w)$ (the length of the time when the user's gaze lives in the bounding box of the token) for each token w

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

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

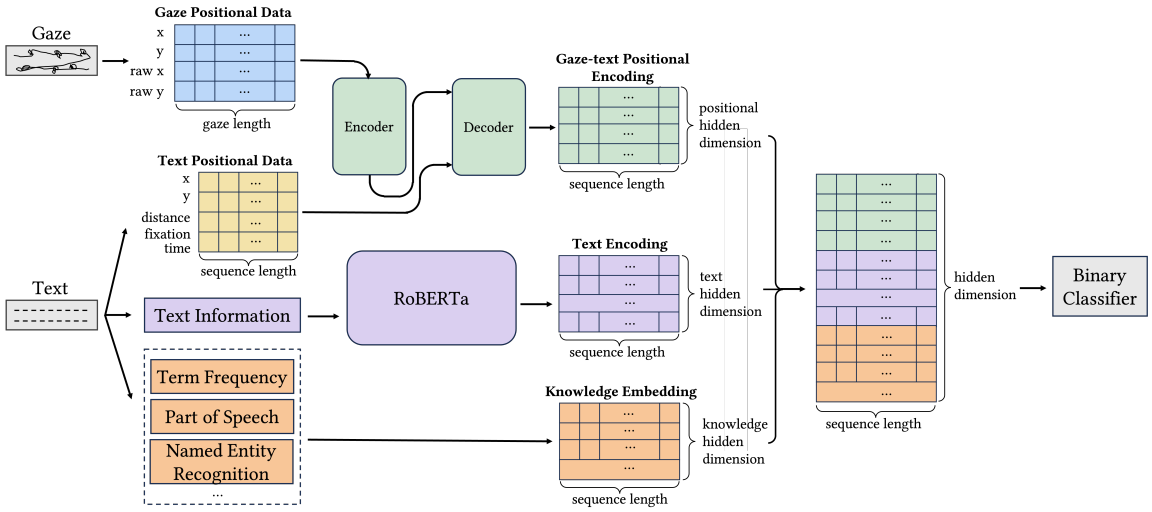

Fig. 4. Our model includes an encoder-decoder model to encode positional data, a pre-trained RoBERTa to encode text information, and learnable embeddings to encode the knowledge. The concatenation of these three matrices is input to a binary classifier.

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, while 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.

3.3.2 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 $s \in \mathbb{R}^{n_{txt}}$ into $Z \in \mathbb{R}^{n_{txt} \times n_r}$, in which the n_r is the hidden dimension of RoBERTa.

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

3.3.3 Knowledge-grounded Enhancement. The pre-trained language model takes the tokens instead of words as the input. The word-level information may be lost during the tokenization. Therefore, we introduce word-level knowledge including the term frequency, part of speech [2], and named entity recognition [11] to utilize word-level information. These features are encoded into a knowledge matrix $K \in \mathbb{R}^{n_{txt} \times n_k}$.

Table 1. Statistics of our collected data.

#Users	#Documents	#Train Data	#Dev Data	#Test Data	#Tokens	#Unknown Tokens
20	120	9802	980	980	380,524	25,233

3.3.4 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 *gamma* are the hyper-parameters that control the weight between the two classes and the speed of the model's focus on difficult examples, respectively.

4 EXPERIMENT

4.1 Main Results

The statistics of our collected dataset are shown in Table 1. We fully fine-tune our model for 30 epochs with the 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 compare our method to three baselines, i.e., heuristic rule based on gaze-word distance, word fixation time, and the logistic regression model trained with these two features above, plus word term frequency, part of speech, and named entity tags.

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 main results of our experiment are shown in Table 2, where we can see that the performance in general, including accuracy, F1-score, precision, and recall of our model trained and tested with data collected with the eye tracker are always better than with webcam. These results highlight the importance of accurate user gaze capturing. Meanwhile, we can see that the performance of our method trained with data collected with the webcam, even though slightly worse compared to our method trained on eye tracker collected data, is still far better than previous methods trained on more accurate eye tracker collected data, which further prove the efficacy and robustness of our method.

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.

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
Webcam	Distance Heuristics	74.1	19.7	12.1	54.0
Webcam	Fixation Heuristics	78.6	21.6	13.8	50.0
Webcam	Logistic Regression	96.7	20.3	13.0	45.6
Eye tracker	Ours	**97.6**	**71.1**	**63.3**	**79.0**
Webcam	Ours	97.3	65.1	60.3	69.7

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

Method	Accuracy	F1	Precision	Recall
Our method	97.6	71.1	63.3	79.0
w/o textual encoding	41.8	22.0	13.5	35.2
w/o gaze encoding	97.5	68.5	63.5	74.4
w/o knowledge embedding	97.6	69.2	68.4	70.0

4.2 Ablation Study

As shown in Table 3, we conduct 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. Firstly, the results indicate that after removing the use of the PLM, our method's performance, regarding all four metrics, significantly decreased with a huge gap, which proves the effectiveness of using PLMs for better capturing the contextual information of the documents in the task of unknown word detection.

Furthermore, our model's performance also has a minimal decrease after removing the gaze information or word-level knowledge. Moreover, we can find that after removing the user's gaze positional information as the input, the model's recall decreases, while its precision remains almost unchanged. This phenomenon implies that gaze patterns, for example, word fixations and gaze-word distances, are helpful for identifying an individual's unknown words based on the user's unique behaviors and patterns. In such cases, it is possible that the model fails to correctly predict the difficulty of a word for the user adaptively because of the lack of user-oriented data.

4.3 Cross-User Generalizability

We evaluate our method with the cross-user generalizability with the leave-one-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 accuracy of our method slightly drops to 97.29%, with the F1 scores, precision, and recall decreasing to 59.6%, 52.8%, and 68.4%. This result also proves the fact that the efficacy of the use of PLMs comes from their internal knowledge of identifying tokens in difficult words, which are generally recognized as unknown by most users. However, over-relying on this factor could weaken our method's performance in cross-user generalization settings.

4.4 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 APPLICATIONS

Many applications can be enabled by our unknown word detection method by acquiring use's unknown words during reading in real time. Foreign language reading can occur in two scenarios. One is literature reading and language learning on 2D interfaces such as laptops or pads. The other is getting information in the surrounding environment through a 3D display such as AR glasses and head-mounted devices. We then discuss the potential applications of our method in these two scenarios.

5.1 Language Learning Assistance

We can divide the functions of language learning assistance into two categories: real-time and non-real-time. By supporting real-time unknown word detection, our method can make translation less obtrusive and help users read more fluently. Our method can track gaze to locate text areas (the sliding window in Fig. 5) and detect unknown words in the area. Then, the application is able to translate these unknown words automatically. It can save users the time of copying and pasting words into the dictionary or retrieving words through the cursor, as well as reducing interruptions to users' reading. At the same time, unknown words can also be automatically added to the user's word list, allowing users to view them at any time.

If the overall reading performance is considered and the real-time feedback is not necessary, many applications can be enabled by summarizing and analyzing the unknown words encountered during the reading process. Potential applications include generating flashcards to facilitate users' memory, counting the user's vocabulary mastery to provide users with learning reports and assessing the forgetting rate to offer users a personalized word learning plan. Combined with generative AI, it is also possible to generate new documents based on recently encountered unknown words. This can help users consolidate vocabulary in an intriguing way.

In summary, fluent reading and efficient word learning are the most urgent needs of second language learners. The highly accurate unknown word detection provided by our method can assist users' language learning in either real-time or summarized manner.

5.2 Reading Assistant in Foreign Language Environment

With the development of augmented reality technology, head-mounted display devices such as Apple Vision Pro[4] will gradually be integrated into daily life in the future. This will allow reading behavior in three-dimensional space to be captured as well. Therefore, we envision that in addition to reading 2D materials, our unknown word detection

[4]https://www.apple.com/apple-vision-pro/

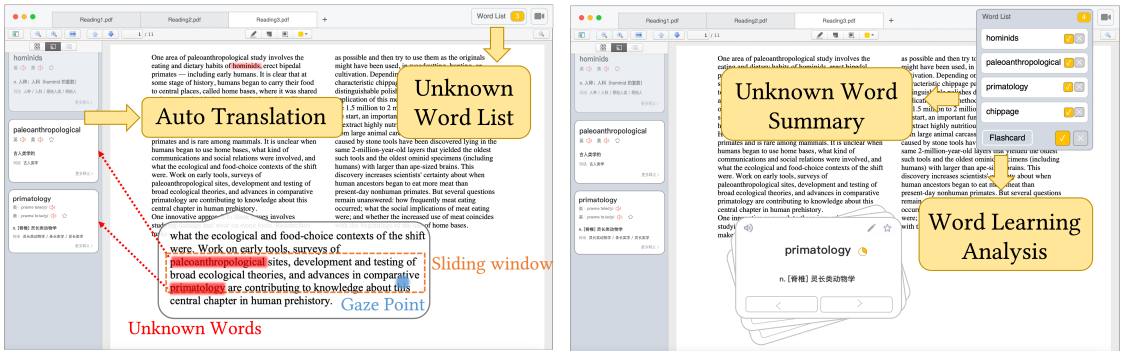

Fig. 5. Applications in 2D language learning scenario: (Left) Real-time auto translation. (Right) Unknown-word summary and word learning analysis.

technology will also be used to assist reading in the three-dimensional world. Moreover, AR headsets are generally equipped with eye trackers, which will enable our method to be easily applied.

Three-dimensional application scenarios include but are not limited to obtaining key information from menus, manuals, and street signs when traveling or living in a foreign language environment and reading commentaries in foreign language exhibitions. Compared with directly displaying large sections of translated text in front of users, providing only key information based on unknown words can reduce the interference to the user's view and reduce the user's burden on extracting key information from a large amount of text. It will provide users with more precise and less intrusive reading assistance in foreign language environments.

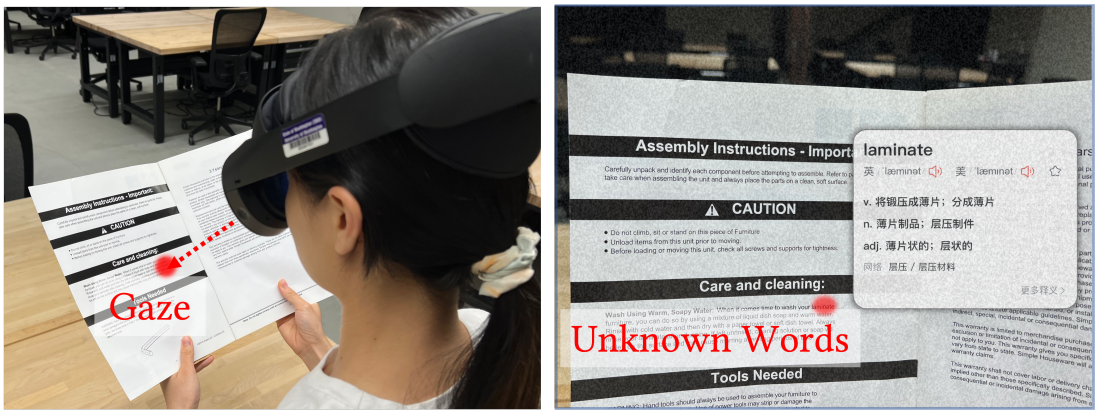

Fig. 6. Applications in 3D AR scenario: (Left) The user wearing the AR headset encounters an unknown word when reading the manual. (Right) The translation of the unknown word pops out automatically.

6 DISCUSSION

6.1 Data Labeling and Preprocessing

Several factors can lead to inaccurate data labeling, thus affecting the performance of the model. The way we label the data has an inherent flaw. In order to reduce the impact of additional operations on gaze behavior, we asked users to

label in the second reading after collecting gaze data in the first reading. However, some users were unable to remember the unknown words they just encountered the second time, resulting in missed labels. Furthermore, different users have different understandings of unknown words. Although we have informed users that unknown words refer to words that they spend extra time thinking about, including words that users take a long time to recall and do not include difficult words that they skip because those words do not affect comprehension, there are still certain differences in the definition of unknown words among users. These two points may directly affect the performance of the model.

There is a contradiction between the sophistication of gaze data preprocessing and the generalizability of the method. Even using a professional eye tracker to collect data, there is still an obvious drift in the data. If we use gaze features such as fixation number and the jump on x direction caused by a newline, the resulting data will be clearer and more beneficial to the results. However, this complex preprocessing will make it difficult to generalize the method to the inaccurate data collected by webcam. That is the reason why we did not perform operations other than the moving average filter and re-sampling on the gaze data. Therefore, our result is less than the best our model can achieve on eye tracker data also caused by this reason.

6.2 Calibration

Calibration is important to obtain high-quality eye-tracking data. But even after the calibration, changes in the user's reading posture and light environment can greatly reduce the data quality. The offset will be changed if the participants move their upper bodies. It will invalidate the previous calibration. Apart from this, reflections from glasses can cause webcam-based eye tracking to freeze temporarily, which will further reduce the quality of the gaze data. Coupled with the drift of the data, eye-movement data contributed only a small part to the overall performance of the model.

For webcam data, we currently use the data of the first page in each session for calibration. This per-session calibration will make our method not real-time in the first article. The inherent inaccuracy of current webcam-based eye-tracking technology causes this issue and it is hard to overcome the influence of users' posture on webcam-based eye tracking. Some technologies can detect the user's distance and head posture relative to the device using sensors on wearable devices [25]. If the existing webcam-based eye-tracking technology can combined with user posture detection, it is possible to reduce the impact of user posture on eye-movement data, thereby reducing the times of calibrations.

6.3 Shortcuts in the Learning of Language Model

Now that PLMs are very powerful and have already learned many high-dimension features of different words. Among them, the level of difficulty could be one of the most significant features. Since the performance drops a lot (41.8% accuracy and 22.0% F1-score) when the textual encoding is ablated, PLM is extremely good at finding unknown words. It may cause shortcuts in the classification of difficult words. However, because there is a large overlap of difficult word set and unknown word set for many users, it is also reasonable that PLM contributes a lot to the improvement of final performance. Despite of the high contribution of PLM, our model is still able to learn from gaze, because the F1-score and recall decrease after the gaze is ablated.

7 CONCLUSION

We propose a high-accurate and real-time unknown-word detecting method that facilitates gaze-based detection using pre-trained language models (PLMs). The text embedding derived from the PLM and the knowledge grounding provide probabilities for candidate words, and the gaze data supply behavior information to the model parallelly. The evaluation shows that it achieves the accuracy of 97.6% and F1-score of 71.1%. The latency of it is within 1 second. Our method

also works on noisy gaze data acquired by webcam, which demonstrates the accessibility and applicability for daily use. After the per-session calibration, the accuracy is 97.3% and the F1-score is 65.1%. A wide range of applications that assist foreign language learning and reading can be enabled by our method. Due to the ubiquity of webcams and the development of AR headsets, we believe that lots of language learners and foreign travelers can ultimately benefit from our technique.

REFERENCES

[1] 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

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

[3] 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

[4] 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

[5] 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.

[6] 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

[7] 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

[8] 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

[9] 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

[10] 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

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

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

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

[14] 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

[15] 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

[16] 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).

[17] 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.

[18] 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

[19] 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.

[20] 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

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

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

[23] 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

[24] 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

[25] 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