HCMUS at the NTCIR-16 RCIR Task

Kim-Nghia Liu*
University of Science, VNUHCM
Viet Nam National University
Ho Chi Minh City, Vietnam
lknghia@selab.hcmus.edu.vn

Vinh Dang*
University of Science, VNUHCM
Viet Nam National University
Ho Chi Minh City, Vietnam
dvinh@selab.hcmus.edu.vn

Thanh-Son Nguyen*
University of Science, VNUHCM
Viet Nam National University
Ho Chi Minh City, Vietnam
ntson@selab.hcmus.edu.vn

Minh-Triet Tran†
University of Science, VNUHCM
John von Neumann Institute, VNUHCM
Viet Nam National University
Ho Chi Minh City, Vietnam
tmtriet@fit.hcmus.edu.vn

ABSTRACT

Reading comprehension provides insightful information on how people read, analyse, and understand text content in various content domains as well as different types of reading behaviors, such as sequential reading, skimming, scanning, or proof reading. In NTCIR-16, the Reading Comprehension in Information Retrieval (RCIR) task provides opportunity for researchers to explore approaches to evaluate comprehension levels of individuals on texts with multi-modal signals, especially with eye-tracking data, and to rank text content in an information retrieval (IR) process with the integration of text comprehension-evidence. Our HCMUS team participate in the Comprehension-evaluation sub-Task (CET) of RCIR task in NTCIR-16. We follow the feature processing and engineering strategy, and adopt different techniques, such as BERT, PCA, and AutoML, to generate output results for this task. Our best approach achieves the Spearman's correlation coefficient of 0.50846.

KEYWORDS

feature processing, feature engineering, AutoML

TEAM NAME

HCMUS

SUBTASKS

Comprehension-Evaluation Task (CET)

1 INTRODUCTION

Reading comprehension of information in text form is a new challenging and interesting field of research, and needs to be explored by scientists. There are various exciting problems related to reading comprehension, such as sorting texts based on their comprehension levels, ranking texts in various topics by integrating text comprehension-evidence into the IR process, *etc.*

Reading Comprehension in Information Retrieval (RCIR [12]) is organized in the NTCIR-16 data challenge as pioneering research on reading comprehension. This pilot task focuses on personalized retrieval techniques that can take advantage of useful information from eye-tracking to ranking text content. In the RCIR dataset, each experimental participant reads several pieces of text and answers 3 multiple choice questions to determine their comprehension of the text. Data are captured from multi-modal sensors to monitor experimental participants in different types of reading behaviors, *i.e.* reading conditions, including **sequential reading** or **skimming** the text to capture as much information as possible, **scanning** through the text to look for a requested information, or **proof-reading** the text to evaluate the number of syntactical or spelling errors.

Participating teams in the RCIR task are expected to propose and develop solutions to utilize multi-modal signals (*e.g.* eye tracking, screenshots, *etc*) in the retrieval process in three sub-tasks. The Comprehension-evaluation sub-task (CET) aims to sort texts based on comprehension levels, while the Comprehension-based Retrieval sub-task (CRT) aims to rank texts by integrating text comprehension-evidence into the IR process. The improvisation ideas to explore the RCIR dataset are encouraged for the Insights (IT) sub-task.

In the RCIR task, HCMUS team participated in the first subtask (CET). Our team's goal is to explore and evaluate various common techniques for addressing the CET problem. We define the common pipeline and strategy for our proposed solutions. Specifically, we first apply data pre-processing techniques to normalize the values of the attributes, or use PCA to reduce the dimensionality of data, as well as select meaningful attributes for information representation. Next, we propose some hand-crafted features or use BERT[5] to encode information of text document in English texts, and propose several representations for the feature vectors. Finally, we use different machine learning techniques to compute the final results, namely Multilayer perceptron (MLP), Random Forest[4], AutoML[8].

Our team propose three methods to realize our strategy, which are presented in Sections 5, 6, and 4. For each particular method, we consider different configurations and evaluate them on the validation set to choose the prominent settings for our official submission. Our best score with the second method achieves the Spearman's rank correlation coefficient metric of **0.50846** on the official private test set.

*Authors contributed equally to the paper
†Corresponding author

The content of this paper is organized as follows. In Section 2, we briefly present existing work related to the RCIR task and solution. In Sections 5, 6, and 4, we present our three methods to solve the CET sub-task of RCIR task. Experimental results are in Section 7. Finally, the conclusion is in Section 8.

2 RELATED WORK

Studying eye movements in reading can provide helpful information about readers and documents. As early as 1998, Rayner presents a study on 20 years of research on eye movements in reading and information processing [15]. Furthermore, eye movements can also provide clues to analyze human behaviors and perception in reading, scene perception, and visual search [16].

Because there can be a noticeable difference in eye movement between skillful and novice readers, it is possible to estimate the language skills of a reader by analyzing eye movement while reading English documents, as in work by Yoshimura *et al.* [20]. The experiments with 11 persons and 10 documents show that the authors can determine the English proficiency level of a person with the accuracy of 90.9% based on their eye movement information.

We can also use the eye gaze to predict a reader's understanding of the content of a document [18]. Experiments are performed with 17 subjects reading 19 documents for a total of 323 recordings. The authors prove that a reader's understanding can be determined more accurately by using eye gaze than by answering questions.

Gaze behavior can also be used to predict reading comprehension [2]. In this work, Ahn *et al.* conduct the experiments with 95 people reading 4 published SAT passages. Tge prediction can be generalized successfully to fixations on new passages from the same readers, and should be further studied to fixations from new readers. Therefore, we think that it would be more appropriate to develop personalized models to adapt to readers' characteristics.

3 COMMON PIPELINE AND STRATEGY

3.1 Common Pipeline

To address the problem in CET sub-task, we analyze all data from the nine volunteers. Each volunteer read 96 texts in 4 different reading conditions, namely sequential reading, skimming, scanning, and proofreading. When a reader read text content, we are provided two main groups of information: eye movement and text data. The eye movement measurements are captured from an eye tracker system while the experimental participant was reading text content. The text data includes the text content together with associated question and answer options that were used to evaluate the reading comprehension of a reader on the text content.

Figure 1: Overview of our proposed pipeline

As mentioned in Section 1, we define our common pipeline and strategy as a guidance to propose different solutions for the CET sub-task, which are realized into three particular methods. In this section and the following two sections, we present our three concrete methods following our defined strategy and pipeline.

There are three main phases in our proposed pipeline, as illustrated in Figure 1, including Pre-processing, Feature Representation, and ML-based Evaluation. The first phase is to normalize data, reduce dimensionality, and select meaningful attributes for the data. The second phase is to propose different feature representations for data, and the last phase is to predict the final result with an appropriate ML-based model for a given input feature vector.

3.2 Common Strategy as Guidance for Our proposed Methods

As the nature of this CET sub-task is how to efficiently exploit valuable information from data captured from multi-modal sensors during the process of human reading text in various reading conditions, we first explore the given data in the RCIR dataset. The data contains 307 original fields/attributes in total. It includes topic ID, text ID, total words of the content, and time reading of participants as well as the score of the tests. Besides, it also includes eye-tracking data. We can easily identify eye-tracking data fields because they have uppercase prefixes. Moreover, they can be classified into several groups. Each group has the same prefix name and has 11 fields corresponding to 11 bins and some other individual fields. Bins fields are named with the lowercase infix "_bin_". In summary, a bin field name in any of these groups has the syntax "<GROUP_NAME>_bin_<index>" where <GROUP_NAME> is the name of the group in uppercase, and <index> is indexed from 0 to 10.

In the **Pre-processing** phase, we normalize the field values to the range $[-1, 1]$ or $[0, 1]$, and we apply PCA[13] to reduce the dimensionality of data. For each of our three proposed methods presented in the following sections, we present our choice for data normalization and dimension reduction.

In the **Feature Representation**, we combine both eye-tracking data (after the pre-processing phase) and text data, including text content and question-answer content. We use BERT[5] as a common utility for text content representation. Besides, as presented in detail in Section 5, we also consider other potential attributes from text data, such as the number of words that are not common English words, or the total length of all questions (in character level), *etc.* We expect that such extra hand-crafted features may provide more valuable information for our solution in the CET sub-task.

Finally, to obtain the final result in the **ML-based Evaluation** phase, we propose to use different ML models, such as Random Forests[4], Multilayer perceptron (MLP), and AutoML[8]. We also employ different techniques for training and optimizing ML models, including Adam [14], SGD [17], Adagrad [7], *etc.*, and several activation layers, such as ReLU [1], Leaky ReLU, Sigmoid, *etc.* From our experiments, the AutoML approaches seem to provide prominent results, as shown in Section 7.

4 METHOD 1 - USING ONLY EYE MOVEMENT DATA AND MULTILAYER PERCEPTRON

In this section, we describe our first method following the proposed common pipeline in Section 3. In this method, we do not use text data in the text content and questions/answers but exploit only data captured from sensors related to experimental participants' activities in reading, especially the eyes information of participants. Our goal is to evaluate if we can estimate the level of reading comprehension only based on eye-tracking data. Our next two methods utilize both eye-tracking data and text information for the CET sub-task.

4.1 Pre-processing

We combine the training and testing sets, then normalize them into range [-1, 1]. We calculate the *max* value in each field of feature vectors normalize the ith component of the feature vector into range $[-1, 1]$ by the following formula: $\frac{x_i}{|max_i|}$ where x_i is the value at the i^{th} component of each feature vector and max_i is the maximum of all values at the i^{th} column.

In this first method, we do not reduce the number of dimensions in data. The dimensionality reduction are adopted in our next two methods.

4.2 Feature Representation

The feature vector is constructed directly from all normalized fields in the raw data of the dataset, and most fields are related to eye-tracking. Observing the data fields for eye-tracking, we argue whether the fields related to "argmax" and "argmin" can provide useful information for our solution. Therefore, we consider two scenarios. We use all data fields of eye-tracking in our first scenario, and ignore all columns with the suffixes "argmax" and "argmin" in our second scenario. In our experiments, we observe that with or without such fields do not have significant difference in results. Thus in our next two methods, we do not remove fields related to "argmax" and "argmin".

4.3 Classification with Multilayer perceptron

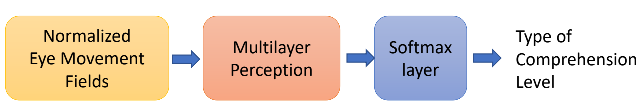

Figure 2: Overview of classification with Multilayer perceptron

We design a multilayer perceptron to train and predict types of comprehension levels end-to-end (see Figure 2). We try several architectures manually to figure out the best one. Finally, we get an architecture as shown in Table 1. Our network contains a Softmax layer at the last step to predict the type of comprehension levels.

We also try several optimizers, such as Adam [14], SGD [17], Adagrad [7], *etc.*, and several activation layers, such as ReLU [1], Leaky ReLU, Sigmoid, *etc.* Finally, the Adagrad optimizer and Sigmoid activation are the best candidates from our experiments.

Table 1: Our Multi layer perceptron architecture

	layer_type(input_dim, output_dim)
layer1	Linear(276, 1024)
layer2	BatchNorm(1024, 1024)
layer3	Sigmoid(1024, 1024)
block4 x 3	Linear(1024, 1024)
	BatchNorm(1024, 1024)
	Sigmoid(1024, 1024)
layer5	Linear(1024, 4)
layer6	Softmax(4,4)

The details implementation and experiments with different configurations are presented in Section 7.1.

5 METHOD 2 - COMBINE BOTH EYE MOVEMENT AND TEXT DATA WITHOUT DEEP FEATURE

In this method, we utilize both eye movement and text data, including the text content as well as questions and answers related to that text. However, we only use traditional techniques, not deep learning methods, to represent data feature and to predict output result. Particularly, we use PCA in the pre-processing phase to reduce the number of dimensions, propose several extra hand-crafted features to represent the complexity of text information, and adopt Random Forest [4] for reading comprehension evaluation.

5.1 Pre-processing

For each record of the data, there are 14 groups of eye-tracking data fields, and each group consists of 11 bins. Thus, we have in total 154 features for eye-tracking data, and consider these bin fields as low-information features. When we apply PCA[13] on these features in the Pre-processing phase, surprisingly, we notice that the first principal component is enough to keep 99.56% information. So we decide to use only one dimension for the PCA feature. Figure 3 shows our method to reduce the dimensions for eye-tracking features.

5.2 Feature Representation

The text content affects the reading comprehension of experimental participants. Furthermore, the complexity of questions and their answer options also contribute to the complexity of the content. Therefore, before using using BERT[5] to encode text in the text content (as in the third method, presented in Section 6), we define some handcrafted features that may provide valuable information related to the complexity of the text content, as well as of the related questions and answers. It means that in our second method, we do not use deep feature to represent both eye movement data and text data.

For text content we define the following extra attributes:

- The number of words that are actual digit numbers. A digit number is an integer or a floating number. A content has many more numbers could be harder to remember and understand. For example it could relate to a scientific topic;

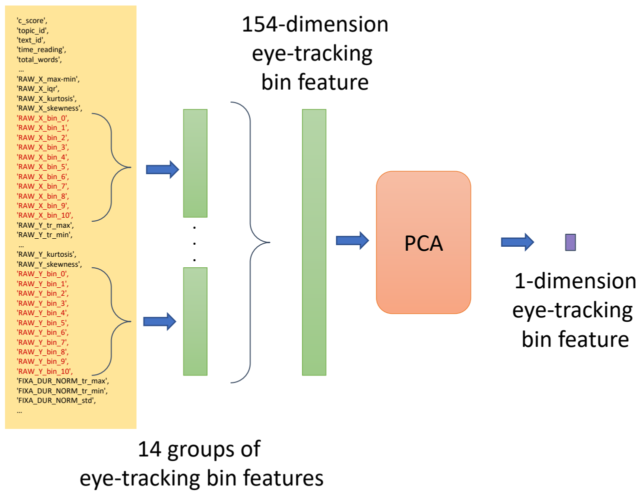

Figure 3: PCA for dimensionality reduction on eye-tracking features

- The number of words that are not common English words. Common English words are defined belong to a set of stop words of NLTK's framework [1] and 5000 common words collected from an internet source; [2]. Note that we already perform WordNet's lemmatizer [3] for nouns, verbs, adjectives, and adverbs for all words before counting them, as our common word set does not cover all forms of a word. This feature can represent the topic's popularity, as content with many rare words could belong to a special topic that not everyone has known. We all know that knowledge and experience of participants greatly contributes to their ability to understand the content. Therefore, the lower popularity of the topic, the larger probability that participants can not clearly understand the content.

For questions and answers, we propose to calculate the following extra attributes:

- The total length of all questions, in character level. This feature could represent the complexity of the question. Because participants' comprehension is estimated by answering these questions correctly, shorter questions could be more straightforward.
- The total length of all options, for all questions, in character level. Similar to the questions, shorter options could be easier for participants to choose the right one.
- The total length of all answers for all questions, in character level. The complexity of the correct answer also affects the question's difficulty.

Questions and answers should recall the text content, so we assume they use the same words as the content. That is why we only focus on their character level rather than counting in word level.

As illustrated in Figure 4, the original remaining eye movement fields, together with one feature obtained by Principal Component

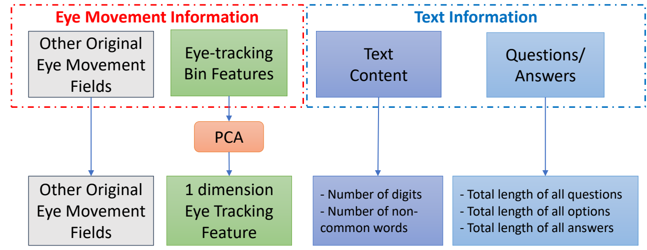

Figure 4: Feature representation in our second method.

Analysis, and the five new text content features, are concatenated in the listed order, for the regression phase to calculate the reading comprehension score of experimental participants. Note that we respect the original order of features when reducing. The remaining Eye Movement features are joined in the same original order, as eye tracking bin features are removed interleaved them.

5.3 Regression for Reading Comprehension Score

In this method, we intend to evaluate if the traditional regression methods can provide good results for the CET sub-task. As we do not normalize or re-scale eye movement data in the pre-processing phase but simply apply PCA on eye-tracking bin features, we think that Random Forest[4] would be an appropriate regression method as it does not rely on a distance algorithm but a tree based method.

Beside Random Forest, we also conduct experiments with Gradient Boosting[9–11] (with different implementations) and AdaBoost [6, 19]. We use Scikit-Learn [3], XGBoost [4] and LGBM [5] libraries. We also perform AutoML[8] on this pre-processed data to let the computer choose the best one. Our experiments on our validation set show that Random Forest method provide the best result for this method. Therefore, we use the Random Forest to train the final model for the private test set. The detail information on evaluating Method 2 is presented in Section 7.2.

6 METHOD 3 - DEEP EMBEDDING AND AUTOML

If Method 2 only uses traditional techniques and hand-crafted features, we intend to explore the application of modern techniques in Method 3. We use BERT to encode text content and combine this feature with the eye movement feature to form the feature vector. We employ AutoML[8] to search for the model configuration that performs the best accuracy on our data.

6.1 Pre-processing

There are 307 fields in eye movement data. Similar to the pre-processing step in Method 2 (see Section 5.1), we focus on 14 groups of eye-tracking bin features, each contains 11 fields. Therefore, we use only these 154 fields to represent eye movement data, and ignore

[1] https://www.nltk.org/
[2] https://github.com/mahsu/IndexingExercise/blob/master/5000-words.txt
[3] https://scikit-learn.org/
[4] https://xgboost.readthedocs.io/
[5] https://lightgbm.readthedocs.io/

other fields, such as "mean", "std", "skewless", "argmax", "argmin", "kurtosis", "min-max", "iqr".

Because the data do not have the same range value, we apply Min-max scaler to normalize values in each field into $[0, 1]$ using the following formula: $X = \frac{X - min(X)}{max(X) - min(X)}$.

6.2 Feature Representation

In this method, we combine two feature vectors into the representation vector, including the eye movement feature vector and the text embedding vector, as shown in Figure 5. After the pre-processing phase, we obtain the eye movement features, each consists of 154 normalized components. For the text embedding vector, we use BERT to represent a text content into a 768-dim vector.

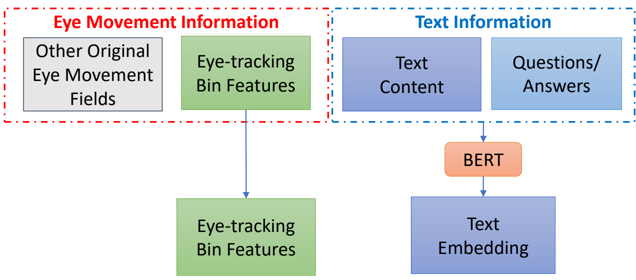

Figure 5: Feature representation in our third method.

6.3 AutoML for Reading Comprehension Prediction

We use AutoML tools from scikit learn to choose the best learning method for the given data. AutoML[8] uses optimization algorithms, such as Bayesian Optimization, to assist us in finding the best model configurations to provide the best results on a specific dataset.

We assume that each person's behaviors on reading and comprehension skills can be different. Thus, we should not combine all data and use only one learning model to learn the behavior of many persons. For Method 3, we do not mix all candidates in the training process. Instead, we split out each candidate to train an adaptive model that can reflect the personalized characteristic of that candidate. For the nine candidates in the dataset, we have nine models.

The detail information on training and evaluating different models in Method 3 are presented in Section 7.3.

7 EXPERIMENTS

7.1 Experiments with Method 1

For this method, we choose the data of the first seven persons for training and the remaining for validation.

We train our model step-by-step to figure out the best settings for Method 1. First, we use ReLU and LeakyReLU for our MLP, and the accuracy is 0.4797 and 0.4703, respectively. Second, we normalize our data to be in the range $[-1, 1]$, we use Sigmoid or Tanh activation functions, but the accuracy is now 0 and 0.1801, respectively. We realize that we should use a Batchnorm layer after each Linear layer and before each Activation layer. The results after applying Batchnorm for the ReLU, Leaky ReLU, Tanh, Sigmoid activation functions are 0.4473, 0.4885, 0.4804, and 0.5368, respectively. We then evaluate different optimizers, including Adam, Adagrad, and SGD. The accuracy corresponding to these optimizers is 0.4676, 0.5368, and 0.5233.

We also consider different numbers of linear blocks with each block corresponding to block4 in Table 1. We use the best parameters of the previous steps to search for the best number of linear blocks. We identify the best option is 3 linear blocks, as presented in Table 1.

7.2 Experiments with Method 2

To train a regression model, we collect all data records from different participants. The validation set is selected randomly from 30% of the training data. For Method 2, we train a universal model for all participants without any information about individuals.

Table 2 reports the result on the same validation set. From our experiment on our validation set (extracted from the training data of RCIR dataset), we decide to train the Random Forest model on the entire training dataset to infer the results for the official test set.

Table 2: Evaluation on the validation set with different regression methods

Method	Result
Scikit-Learn's Linear Regression	0.52228
XGBoost	0.62309
Scikit-Learn's AdaBoost (Decision Tree)	0.63222
Scikit-Learn's Gradient Boosting	0.64639
AutoML	0.64888
LGBM	0.66355
Scikit-Learn's Random Forest	**0.69801**

7.3 Experiments with Method 3

For Method 3, we also train our candidate models with 70% of training data, and use 30% of training data for validation. We also consider the training strategy similar to Method 2. It means that we train a universal model that can be used to predict reading comprehension for all persons. However, through our experiments on the validation set (extracted from the training set of RCIR dataset), we notice that a personalized model can provide better results than a universal model. Therefore, we decide to train different models for each reader.

We also try to use PCA to reduce the number of dimensions for eye movement features, or the combination of eye movement and BERT features. However, the obtained results on the validation set is not promising. Therefore, we come to the solution with the specific settings as presented in Section 6.

7.4 Evaluation on Official Test Set

In the previous three sub-sections, we present our step-by-step experiments locally with the validation set extracted from the training set of the RCIR dataset. We choose the most promising configurations for each of our proposed methods from the experiments to train the models on the whole training set. Then we use our best models of the three methods to predict the results for the test set of the RCIR dataset in the CET sub-task.

Table 3 shows the results (Spearmans correlation coefficient) of our three methods on the test set of RCIR dataset for the CET sub-task. Based on the evaluation on the test set of RCIR task, the Spearmans correlation coefficient of our proposed methods 1, 2, and 3 are 0.4024, 0.49182, and 0.50846 respectively.

It is obvious that using only eye movement information (as in Method 1) cannot provide the results as good as combining both eye movement and text information (as in Methods 1 and 2). When we utilize text information, the results can be boosted significantly from 0.4024 to 0.49182 and 0.50846. Besides, in Method 2, we only extract some extra hand-crafted attributes from text information, and the result of Method 2 is slightly lower than that of Method 3, which employs BERT to encode the whole text into feature vectors.

Table 3: Results on official test set

Method	Result
Method 1	0.40242
Method 2	0.49182
Method 3	0.50846

8 CONCLUSIONS

This paper presents three methods to solve the Comprehension-evaluation Task (CET) in the Reading Comprehension in Information Retrieval (RCIR) challenge in NTCIR-16.

In our methods, we use different data pre-processing and feature engineering techniques. We use only eye movement data in our first method, while we use both eye movement and text information for the other two methods. Experiments on the test set of the RCIR dataset show that utilizing both information can provide better results in evaluating reading comprehension for readers. We use BERT to encode text information. We also define several helpful attributes related to the complexity of text documents and questions and answers to provide more information for our models.

We use Multilayer Perceptron, Random Forest, and AutoML to estimate the reading comprehension of readers. Our best solution achieves the Spearmans correlation coefficient of 0.50846 in the official test set of the CET sub-task in RCIR challenge 2022.

As the RCIR task is a new and interesting one, we aim to study different techniques further to boost the results for text comprehension evaluation by taking advantage of helpful information from eye-tracking systems.

ACKNOWLEDGMENT

This work was funded by Gia Lam Urban Development and Investment Company Limited, Vingroup and supported by Vingroup Innovation Foundation (VINIF) under project code VINIF.2019.DA19.

REFERENCES

[1] Abien Fred Agarap. 2018. Deep Learning using Rectified Linear Units (ReLU). http://arxiv.org/abs/1803.08375 cite arxiv:1803.08375Comment: 7 pages, 11 figures, 9 tables.

[2] Seoyoung Ahn, Conor Kelton, Aruna Balasubramanian, and Greg Zelinsky. 2020. Towards Predicting Reading Comprehension From Gaze Behavior. In *ACM Symposium on Eye Tracking Research and Applications* (Stuttgart, Germany) *(ETRA '20 Short Papers)*. Association for Computing Machinery, New York, NY, USA, Article 32, 5 pages. https://doi.org/10.1145/3379156.3391335

[3] Steven Bird and Edward Loper. 2001. Natural Language Toolkit: WordNet stemmer interface. https://www.nltk.org/_modules/nltk/stem/wordnet.html.

[4] Leo Breiman. 2001. Random Forests. (2001), 5–32.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. http://arxiv.org/abs/1810.04805 cite arxiv:1810.04805Comment: 13 pages.

[6] Harris Drucker. 1997. Improving Regressors Using Boosting Techniques. *Proceedings of the 14th International Conference on Machine Learning* (1997).

[7] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12, Jul (2011), 2121–2159.

[8] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. 2015. Efficient and Robust Automated Machine Learning. In *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2015/file/11d0e6287202fced83f79975ec59a3a6-Paper.pdf

[9] Jerome Friedman. 2001. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics* 29 (10 2001), 1189–1232. https://doi.org/10.2307/2699986

[10] Jerome Friedman. 2002. Stochastic Gradient Boosting. *Computational Statistics Data Analysis* 38 (02 2002), 367–378. https://doi.org/10.1016/S0167-9473(01)00065-2

[11] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2001. The Elements Of Statistical Learning. *Aug, Springer* 1 (01 2001). https://doi.org/10.1007/978-0-387-21606-5_7

[12] Graham Healy, Tu-Khiem Le, Minh-Triet Tran, Thanh-Binh Nguyen, Boi Mai Quach, and Cathal Gurrin. 2022. Overview of the NTCIR-16 RCIR Task. In *Proceedings of the 16th NTCIR Conference on Evaluation of Information Access Technologies (NTCIR-16)*. Tokyo, Japan.

[13] Harold Hotelling. 1933. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* 24 (1933), 498–520.

[14] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. http://arxiv.org/abs/1412.6980 cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[15] Keith Rayner. 1998. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin* 124 3 (1998), 372–422.

[16] Keith Rayner. 2009. The 35th Sir Frederick Bartlett Lecture: Eye movements and attention in reading, scene perception, and visual search. *Quarterly Journal of Experimental Psychology* 62 (2009), 1457 – 1506.

[17] Sebastian Ruder. 2016. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747* (2016).

[18] Charles Lima Sanches, Olivier Augereau, and Koichi Kise. 2017. Using the Eye Gaze to Predict Document Reading Subjective Understanding. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Vol. 08. 28–31. https://doi.org/10.1109/ICDAR.2017.377

[19] YoavFreund and Robert ESchapire. 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. System Sci.* (1997), 119–139.

[20] Kazuyo Yoshimura, Koichi Kise, and Kai Kunze. 2015. The eye as the window of the language ability: Estimation of English skills by analyzing eye movement while reading documents. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. 251–255. https://doi.org/10.1109/ICDAR.2015.7333762