DCU Team at the NTCIR-16 RCIR Task

Manh-Duy Nguyen
Dublin City University
Dublin, Ireland
manh.nguyen5@mail.dcu.ie

Thao-Nhu Nguyen
Dublin City University
Dublin, Ireland
thaonhu.nguyen24@mail.dcu.ie

Binh Thanh Nguyen
AISIA Research Lab
VNU HCM - University of Science
Ho Chi Minh, Vietnam
ngtbinh@hcmus.edu.vn

Annalina Caputo
Dublin City University
Dublin, Ireland
annalina.caputo@dcu.ie

Cathal Gurrin
Dublin City University
Dublin, Ireland
cathal.gurrin@dcu.ie

ABSTRACT

Reading is one of the most common everyday activities. People read through most of their daily context such as during study or for entertainment in their spare time. Despite playing a critical role in our lives, there has been limited research on how people read and how it affects their level of understanding. The NTCIR-16 RCIR challenge is the first collaborative evaluation that aims to automatically measure the reading comprehension of a reader and integrate it as part of the information retrieval process. In this paper, we present our approach for the NTCIR-16 RCIR challenge, in which task participants are required to predict reading comprehension using eye movement signals of the readers. We utilised several conventional machine learning techniques to estimate the level of comprehension and combined it with a language model to perform text retrieval. Our extensive experiments, covering both subject-dependent and subject-independent scenarios, showed that our approach with fine-tuning obtained a Spearman's coefficient of 0.5993 for the comprehension-evaluation task and nDCG at 0.7296 for the comprehension-based retrieval task.

KEYWORDS

reading comprehension, machine learning, language modelling

TEAM NAME

DCU

SUBTASKS

Comprehension-evaluation task (CET)
Comprehension-based retrieval task (CRT)

1 INTRODUCTION

Reading plays a critical part in our daily lives. We spend most of our time reading, since texts are present in many places surrounding us, such as on the screen of mobile devices or in advertisements. However, we still have limited knowledge about how people read and how it affects their comprehension. There is some research in this field, where eye movements are assumed to have a key role in reading tasks [3, 4, 7, 8]. For instance, people tend to have different eye movements when they read text with difficult concepts that are not related to their background knowledge[9, 15]. Therefore, some works have been proposed to predict reading comprehension based on eye-tracking signals [2, 14, 18]. The results, although needing improvements, have opened a wide variety of applications for using reading comprehension. It can be applied in the education area where a prediction model could give instant feedback to assess the level of comprehension of teaching documents. In addition, it can help readers avoid time-consuming self-evaluation surveys about their understanding regarding the text they have read. Nevertheless, employing reading comprehension to facilitate personal text retrieval is still an open question.

RCIR (Reading Comprehension in Information Retrieval) [6] is a pilot challenge task in the NTCIR-16 conference. This challenge aims to use the reading comprehension level as an input to facilitate text retrieval. Therefore, there are 2 subtasks in the challenge, which are the Comprehension-evaluation task (CET) and the Comprehension-based retrieval task (CRT). Regarding the former subtask, participants need to build a model to predict the level of comprehension of the reader based on their gaze behaviour. The predicted values are also integrated into the text retrieval model for the latter subtask, in which task participants are required to find a text relevant to a given topic using the comprehension level.

In this research, we explore various conventional machine learning techniques to address the CET subtask. A feature selection stage was applied to select the useful features. We ran the experiments with different scenarios to have a deeper insight into the performance of our model. Furthermore, the content of the text was also taken into account in our regression model to validate how this semantic information contributes to it. In terms of the CRT subtask, we firstly applied a SBERT [16] language model to find relevant texts. The estimated reading comprehension values were then integrated to measure the similarity score to produce the final ranked list.

2 RELATED WORK

Some approaches have been introduced to predict the level of reading comprehension. The discriminant function analysis model proposed by Underwood et al. was seen as a pioneer work [18]. In their research, the authors employed a list of features related to eye fixations to estimate the comprehension level, then used it to classify whether a reader belongs to a higher skilled reading group or a less skilled reading group. Their results showed that fixation duration had a strong correlation with reading comprehension, but vocabulary in the text and readers' reading speed were not a meaningful feature.

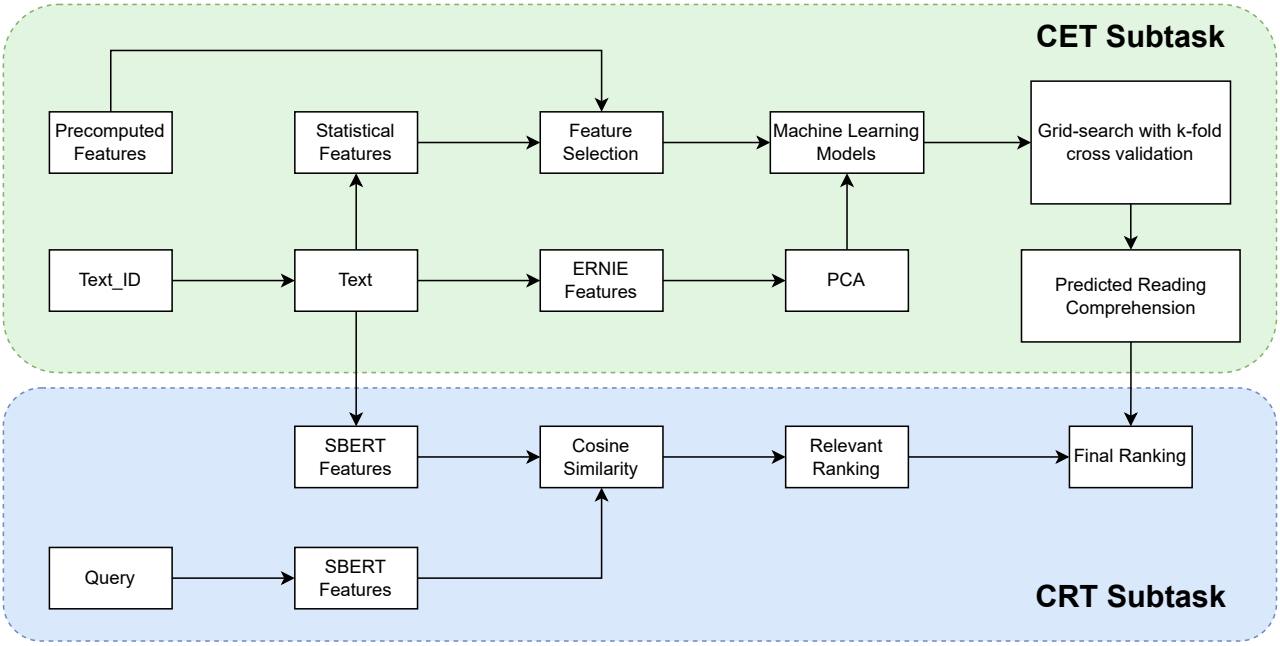

Figure 1: Our pipeline for CET and CRT Subtasks

With another point of view, Makowski et al. [14] considered eye movements during reading as personal biometrics. Therefore, they developed a support vector machine (SVM) model that can identify readers based on their gaze behaviour and the lexical features extracted from the reading text. These semantic features could be the frequency and length of a word in the text, the first word of a sentence, or the binary tag indicating if that word was a jargon term of a specific subject area. The model was also applied to estimate the levels of text understanding using the same features. Although they found that eye movement was a good predictor to identify readers, their features were not reliable to predict reading comprehension.

Recently, deep learning models have been applied in this research area. Convolutional neural networks (CNN) and recurrent neural networks (RNNs) were used to predict reading comprehension from gaze signals, including fixation duration, saccade amplitude, pupil size, and reading rate [2]. However, their finding was that the deep learning models created no impact on the accuracy compared with the baseline, which was a simple majority vote classifier, due to the label's inconsistency in the dataset.

Although having the same motivation of predicting reading comprehension, the RCIR challenge provided their own unique dataset that is different from those used in the aforementioned works. The dataset includes not only the pre-computed features from eye-tracking signals but also information related to the text such as reading time, number of words in the text, and even the entire content of the text as well. While other existing datasets consider predicting comprehension as a binary classification problem, the labels in RCIR dataset consist of 4 different categories. This intricate dataset can inspire participants in their analysis to gain more insight into the contribution of these features to the comprehension prediction problem. Furthermore, it is worth mentioning that the CRT subtask of RCIR is the first challenge that requires participants to develop a personal text retrieval model that utilises reading comprehension measures.

In this work, our contribution is threefold. Firstly, we address the CET task by introducing our framework to estimate reading comprehension. The pipeline includes the feature selection procedure at the beginning, and a grid search to select the most appropriate machine learning model. Fine-tuning of the hyperparameters is done using the k-fold cross-validation technique. Moreover, we also investigated whether the content of the text affects the reader's understanding by using the text-encoded features extracted by the ERNIE model [17] in addition to the eye movement features. Secondly, we present our comprehension-based text retrieval model to solve the CRT subtask. The ranked list was created by combining the reading understanding scores predicted in the CET subtask with the similarity scores between a given topic with texts in the dataset produced by the SBERT model [16]. Thirdly, our experiments shed light on the performance of our proposed model in regards to subject-dependent and subject-independent cases. Moreover, the contribution of the number of features and the semantic information of the context of the texts are also assessed.

3 DATASET

The RCIR challenge provided a detailed reading dataset that consisted of the pre-computed features from the eye-tracking signals of 9 volunteers. Each participant was required to read 24 pieces of text for each reading condition including reading, skimming, scanning, and proofreading. As a result, each volunteer had to read 96 texts in total, in which 24 texts were shared with all other readers, 24 texts were similar with another reader, and 48 texts were unique to that volunteer. From these 96 texts, there were 72 texts provided

in the training set and 24 unique texts were used for testing for each volunteer. Each text followed RACE dataset [12] topics such as transportation, art, or health, etc. It is noted that the number of topics was similar across all reading conditions, and the topics in the test set were different and did not appear in the training set. In addition to the labeled reading comprehensive score (an integer ranging from 0 to 3), there were 306 features in the dataset that included the text identifier (to get the content of the text), number of words in a text, the topic of the text, the reading time of a volunteer for that text, and 302 pre-calculated features of their gaze behaviour. Moreover, the users' identity (already anonymised) and the content of the texts were also included in the dataset. Regarding the CET subtask, participating teams were asked to estimate the comprehension level based on the given features. The evaluation metric was Spearman's rank correlation coefficient. The CRT subtask required teams to retrieve the texts that were relevant to a given query and sort by comprehension level, such as *"Find the texts that talk about animals in general"*. Nevertheless, the similarity score, which measured how relevance between the text and the query was, took the level of reading comprehension of the text into account.

4 METHODS

Deep learning models have achieved state-of-the-art results in many research areas. However, one of the critical requirements of using this technique is having a large number of training data samples because of the massive number of parameters in the network. The labeled dataset used in RCIR challenge only consisted of 648 samples (9 volunteers and each read 72 texts in the training set) coming with 305 features. We considered that this number of training samples was not sufficient for a deep learning approach. Therefore, we decided to apply conventional machine learning models for the challenge. Figure 1 illustrates our approach for both CET and CRT subtasks.

4.1 CET Subtask

The evaluation metric for this subtask is the Spearman's correlation coefficient, which is often applied in regression problems. We chose a list of common regression models for this subtask, including Linear Regression, Random Forest Regressor, Gradient Boosting Regressor, AdaBoost Regressor, and Epsilon-Support Vector Regressor. It is important to note that the labels for reading comprehension in the dataset were all integers ranging from 0 to 3. This is a suitable data type to apply a classification model. Therefore, we also included the classification form of the above-mentioned techniques in our experiments. However, instead of predicting the class as usual, the output of a classifier now would be converted into a float number to be evaluated with Spearman's coefficient by the following formula:

$$p = \sum_{c=0}^{3} p_c * c = \sum_{c=1}^{3} p_c * c$$

where p is the estimated reading comprehension, p_c is the output of the classifier indicating the probability of a sample being classified as class c. In total, our list of models for the CET subtask consisted of 5 regression models and 4 classification models (Random Forest Classifier, Gradient Boosting Classifier, AdaBoost Classifier, and C-Support Vector Classifier). All 9 models are supported in the scikit-learn library[1].

Regarding features, we extracted more features that were related to the content of the reading texts. Specifically, for each sample in the dataset, we firstly find the content of a text based on its identifier. We then used the Spacy model[2] to calculate some simple statistical features of the text such as counting the number of nouns, verbs, adjectives, or entities, etc. This stage extended the dataset by 36 features. Moreover, we wanted to integrate the features that could capture the meaning of the text to our models. We, hence, employed ERNIE [17] to encode the text into a 768-dimensional feature. Regarding the paragraph that contained more words than the ERNIE configuration, we split it into chunks that were suitable with the setting. The embedded features of the text now were the average of the features of chunks. On the other hand, we discarded the topic and text identifiers of texts out of our models. We think that the topics would be the same for many texts, hence this feature was too general and not meaningful to distinguish reading comprehension scores. Meanwhile, the text identifiers were just to help us to get the content of texts but did not contain any information itself. In summary, there were 1108 features including 304 provided features (topics and text identifiers were excluded), 36 statistical features, and 768-dimensional features from ERNIE. All features were normalised in the preprocessing stage.

The number of samples in the dataset was extraordinarily low compared to the number of features, which was 648 samples with 1108 features. This might lead our models to suffer from the curse of dimensionality problem which can results in a machine learning model producing unreliable results. It is in demand to have a feature selection technique to overcome this phenomenon. We adopted the approach introduced by Li et. al. [13] to reduce the number of features in our experiments. Specifically, the importance of a feature was calculated with the following methods:

- **Chi2**. With the idea that a feature with a higher variance contains more useful information to distinguish between labels, we rank the feature importance based on their variance. Features with higher variance will be more important.
- **GBDT**. Gradient Boosted Decision Tree (GBDT) [10] is not only a classification model but also a feature selection technique [19]. We applied this approach to calculate the feature importance based on which we can rank the features.
- **Correlation**. We measured the correlation between each feature and the true labels. If they are strongly correlated, which is to have a high correlation in either a negative or positive manner, the features will have higher importance than others. In addition, since there were many features in our dataset, there was a high chance that 2 features were correlated with each other. This means that they might contain similar information and even can be harmful to the performance of a model. We found pairs of correlated features and lowered the importance ranks of those having a lower correlation with the labels.

[1] https://scikit-learn.org/stable/supervised_learning.html
[2] https://spacy.io/usage/models

Table 1: Spearman's coefficient of models for the validation set in 3 scenarios with different numbers of selected features. The number in bold is the highest score in that row.

Scenario	#Features	#PCA	ADC	ADR	GBC	GBR	LNR	RFC	RFR	SVC	SVR
SI	0.5	N/A	0.3195	0.3769	0.3989	**0.4658**	0.2507	0.3223	0.3425	0.3587	0.2187
	1	N/A	0.163	0.3301	0.306	**0.4159**	0.1853	0.2759	0.3116	0.3279	0.16
	0.5	150	0.2935	0.3906	0.3521	**0.4207**	0.0887	0.29	0.3262	0.3229	0.0307
	1	150	0.2819	0.3856	**0.4195**	0.4099	-0.0053	0.2532	0.3102	0.2568	0.0368
	1	768	0.1237	**0.4133**	0.2818	0.4054	0.1026	0.2207	0.297	0.1197	0.1243
SD	0.5	N/A	**0.5864**	0.5274	0.3982	0.3539	0.1519	0.5608	0.5304	0.5696	0.4272
	1	N/A	0.4739	0.5006	0.3773	0.366	0.282	**0.5527**	0.5276	0.5182	0.4039
	0.5	150	0.5456	0.4558	0.379	0.5056	0.4147	0.5556	0.536	**0.5784**	0.4898
	1	150	0.4504	0.5267	0.4843	0.3618	0.2731	**0.5776**	0.5287	0.5182	0.4224
	1	768	0.5072	0.4526	0.4037	0.2838	0.3098	**0.517**	0.5062	0.4526	0.4124
GE	0.5	N/A	0.4889	0.5544	0.5816	**0.5846**	0.435	0.5015	0.526	0.5368	0.5444
	1	N/A	0.4712	0.5676	0.5563	**0.5854**	0.3318	0.4995	0.5257	0.4596	0.4834
	0.5	150	0.476	0.5588	0.5779	**0.6148**	0.3574	0.5269	0.5429	0.5269	0.5258
	1	150	0.462	0.5661	0.5646	**0.6003**	0.2873	0.5095	0.5411	0.5011	0.4864
	1	768	0.4425	0.5336	0.5407	**0.5956**	0.077	0.5023	0.5392	0.4553	0.4352

- **PCA**. Principal Component Analysis (PCA) [1] is the method that can reduce the size of features while retaining their trends and patterns [11]. Because ERNIE-extracted features contained 768 elements that need to come together to express the encoded information of a text, we could not apply the methods described above to decrease its size. We used PCA to project these high-dimensional features to a lower dimensional space.

We calculated the importance of the non-ERNIE features using the Chi2, GBDT, and correlation approaches. We then ranked them in regard to each approach. The final ranking, and also the feature importance, was the mean of the respective feature ranking of the 3 ranked lists.

4.2 CRT Subtask

Because this was a retrieval task, the most important step was to measure the relevance between the texts' contents and the queries. To calculate the similarity scores, we initially utilised SBERT [16] to encode each sentence of a text into a 384-dimensional feature. The same was true for a query. The relevance of the text and the query was the average of the cosine similarity of the feature of the query with the feature of each sentence in the text. After getting the content-based relevance scores, we used the following formula to integrate the reading comprehension into the retrieval. The modified similarity score was calculated as follows:

$$sim(t, q) = (P(t) + 1) * R(t, q)$$

where $sim(t, q)$ is the similarity score of the text t and the query q, $P(t)$ is the estimated comprehension level calculated from the CET subtask, and $R(t, q)$ is the cosine similarity of t and q.

After investigating 6 queries in this subtask, we found that each query was about a specific topic in the test set. For example, query 2 was about retrieving all texts related to animals in general, which matched with topic 7 in the testing data. According to the guidelines from the organiser, each volunteer had to read 4 texts for each topic in the test set. Therefore, we need to rearrange the retrieved list of texts created based on $sim(t, q)$ similarity score to maximise the evaluation metric. Regarding one query, we extracted the top-m ($m = 4$) most relevant texts read by each volunteer and put them on the top of the list. The remaining of the list was from the original result.

5 EXPERIMENTS

5.1 CET Subtask

We conducted our experiments in 3 different ML setups that includes subject-independent (**SI**), subject-dependent (**SD**), and general (**GE**). The SI scheme predicted the reading comprehension score for a reader without using their samples in the training stage but other readers. In contrast to the SI scenario, SD estimated a reader's comprehensive score based on the samples belonging to that reader only. Meanwhile, the GE setup combined SI and SD schemes where we used samples of all readers to predict a reader's comprehension.

There are many hyperparameters in conventional machine learning models that can have a huge impact on the performance of the models. Typical examples of hyperparameters are the number of trees in Random Forest, the learning rate in the Gradient Boosting Tree, or the kernel function in Support Vector Machine. It is, hence, critical to select the appropriate values for these hyperparameters. To address this issue, we performed a grid-search tuning technique to find the optimal configuration for each model. In addition, due to the limited number of training samples, the k-fold cross-validation method ($k = 5$) was employed to evaluate the performance models with their hyperparameters option. It is noted that we only applied this validation method in SD and GE scenarios and the size of a fold was set at 20% of the entire training dataset.

5.2 CRT Subtask

We ran this subtask after we found the best prediction of reading comprehension on the test set that was evaluated by the organiser. We applied the estimated comprehension to revise the ranked list of

relevant texts that were measured by the cosine similarity between SBERT-encoded features of texts and queries. Furthermore, the impact of different model structures of SBERT (**Fast-Mini**, **Mini**, and **Base**)[3] on the retrieval results were also investigated. The Fast-Mini and Mini versions of SBERT produce a 384-dimensional vector from a text while the Base model encodes the text to a 768-dimensional vector. The Fast-Mini structure, although having the same outcomes' size as Mini, contained only half of the transformer layers in the model compared to that of Mini version.

6 RESULTS AND DISCUSSION

6.1 CET Subtask

Table 1 shows Spearman's coefficient (r) on the validation set for 3 setups that were SI, SD, and GE. For each setup, we tried different combinations of a portion of non-ERNIE features (340 features) and the number of the projected components (from the PCA) in terms of the ERNIE features (768 features). For instance, the first row was the result when we only used #Features = 50% of the non-ERNIE features, which were 170 features, and did not use any features from the ERNIE features (#PCA = *'N/A'*). It is noted that the results reported in Table 1 were all obtained after running k-fold cross-validation ($k = 5$) with the grid-search strategy to find the optimal configuration. Furthermore, the validation sets across the scenarios were different but similar between the machine learning models within the same scenario. For example, the validation set in SI was all samples from specific subjects, whilst that in SD and GE were the stratified sampling from the training set to ensure the balance in the label distribution between training and validation set. Consequently, the results across scenarios were not suitable for comparison due to its different validation set but only comparison in the performance of models within the same scenario.

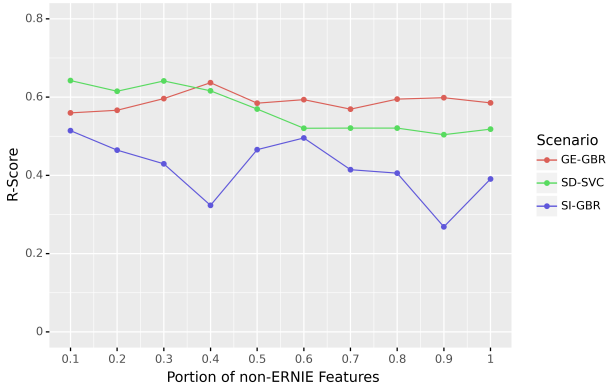

Figure 2: R-Scores of 3 scenarios with different number of selected features (validation set).

We observed that the GBR model worked better in the SI and GE scenarios, while RFC performed best in the SD scenario. Given this, we then employed these models with the optimal hyperparameters to predict reading comprehension in the testing set later. As can be seen in the first 2 rows of each scenario, it seemed that the models tended to achieve better r-scores by reducing half of the features. It has shown the effectiveness of the feature selection stage when addressing the dataset with an enormous number of features. Regarding ERNIE features, the greater the number of them included in the models, the worse the prediction produced by the models. For instance, on lines 1, 3, and 5 of the SI scenario, GBR obtained a score of 0.4159 when using all non-ERNIE features. However, its scores reduced to 0.4099 and 0.4054 when there were more PCA-based features included in the model. It meant that the content of the reading texts did not contribute much to the model performance. It can indicate that the comprehension level of subjects might not be dependent on the paragraphs they had read but on their biosignals. There was another reason behind this finding. This research encoded the entire text by taking the average of embedded of each chunk of the text. This approach can be a problem for a long paragraph where the detail information would be generalised to noise. Therefore, based on Table 1, we decided to discard ERNIE features from further experiments.

Figure 2 showed the change in the r-scores in the validation set for each scenario in terms of increasing the portion of selected features. Here, we used the GBR model for the SI and GE scenarios while the SVC model was employed in the SD scenario. The SVC was chosen since it can work with a small sample size [5] while tree-based models usually overfit with the small dataset. Regarding all 3 scenarios, using entire non-ERNIE features did not guarantee the highest scores. There was a decrease in the r-scores of SD-SVC and SI-GBR when increased' the percentage of features. Regarding GE-GBR, there were no significant differences in performance in terms of the portion of the features, but the use of 40% gained the best score.

Table 2: Spearman's coefficient in testing set.

RUN_ID	Scenario	#Features	#PCA	R-Score
0	SI-GBR	0.5	N/A	0.4038
8	SI-GBR	0.6	N/A	0.3389
3	SD-SVC	0.1	N/A	0.5119
4	SD-SVC	0.3	N/A	0.5992
2	SD-SVC	0.5	N/A	0.5600
5	GE-GBR	0.4	N/A	0.5165
1	GE-GBR	0.5	N/A	0.5529
6	GE-GBR	0.5	150	0.5232
7	Combine	-	-	0.6000

Based on the result in Figure 2, we selected some of the appropriate settings to apply the trained models on the testing set. Table 2 shows the Spearman coefficient in the testing set with different settings. In the testing set, we used the model trained on a specific reader to predict their reading comprehension regarding SD and SI scenarios. Regarding GE scenario, since it was not dependent on the subject identifier, we applied the same model to estimate the comprehension level of all volunteers. We observed that the SVC model trained SD scenario obtained the highest r-score compared to other scenarios. Specifically, the SD-SVC using only 30% features achieved a score o 0.5992 (RUN_ID 4), which was higher than that of SI-GBR and GE-GBR (RUN_ID 0 and RUN_ID 1, respectively).

[3]https://www.sbert.net/docs/pretrained_models.html

Table 3: Example of Keywords Type used in query 2 in the CRT subtask

Keywords Type	Keywords
T1	animals with their life, habit, abilities, benefit and endangerment.
T2	animals and animals habit and endangerment.
T3	animals, animals habit.
T4	animals, elephants, wild, zoo.

This was because the model that trained only with samples from a specific volunteer would not be affected by noise from other volunteers.

In the regard of each scenario, the results in the validation set did not reflect the scores in the testing set. For example, as shown in Figure 2, SI-GBR with 50% selected features performed worse than it with the help of 60% features in the validation set. In contrast to that finding, the results on the testing set were in the opposite. Using half of the features increased the r-scores by 7%. The same was true in RUN_ID 2-3-4 and RUN_ID 5-1. The reason for the differences in the results in the 2 sets was the number of training samples. It was hard that the models can learn and capture entirely the patterns in the sample with the limited number of training data. We also tried including ERNIE features for the testing set. Nevertheless, the r-score decreased accordingly from 0.5529 to 0.5233 in RUN_ID 1 and 6. We combined the result in RUN_ID 0, 4, and 1 to produce RUN_ID 7. The estimated reading comprehension now was the softmax-weighted sum of the 3 models for 3 different scenarios. The weights in this submission were the softmax of the R-Score returned from the organiser. The result was slightly increased and was not significant.

6.2 CRT Subtask

In this subtask, we used the prediction from RUN_ID 4 of the previous task. Although RUN_ID 7 got the best performance, it was based on the scores returned by the organiser. Therefore, we did not consider its result for the CRT subtask. We note that we did not train any model for this subtask.

In the experiment for CRT subtask, we tried different keywords types to retrieve texts based on the given queries as provided by the organiser. The keywords in **T1** were almost similar to the queries, while **T2** was the short version of T1 where we removed some redundant words. **T3** was similar to T2, but only contained keywords that were nouns. **T4** used the most frequent words, which were mentioned in the guidelines, in the test set as keywords. The example of keywords is shown in Table 3.

Table 4 illustrates the nDCG scores in different settings. The "*SBERT Type*" column refers to the structure of the SBERT model used to measure the similarity between a query and the content of a text. The "*top-m*" column indicates the top m relevant texts that correspond to the query, as mentioned in Section 4.2. The "*Keywords Type*" column shows the type of keywords that was used to solve queries.

We first wanted to evaluate the initial ranked retrieved texts without using reading comprehension scores but based on SBERT encoding only (RUN_ID 1). The nDCG was roughly at 0.5856 which could be improved to 0.6929 if the results of the CET subtask were added as shown in RUN_ID 0. Although each volunteer read 4 texts on the same topic, we extended the top-m in case the retrieval model missed some texts. We found that using $m = 6$ produced a better nDCG score than other options from RUN_ID 0 and 2-5. Different structures of the SBERT model were also taken into account. RUN_ID 3, 6, and 8 revealed that the models with more layers in their structure gained lower scores. The Fast-Mini SBERT model obtained the best result at 0.7245 (RUN_ID 3) compared to other structures. Regarding the keywords type, T3 achieved the nDCG at 0.7289 (RUN_ID 10), which was higher than that of the remaining types. This can be because T3 did not contain unnecessary words likes T1 and T2. Furthermore, the keywords in T3 were also relevant to the queries whilst those in T4 were just the common words in the topics and might not be mentioned in either queries or texts. For example, the word "*elephants*" (T4), which was one of the most common concepts in the animal topic, was used to solve query 2 which was also about the animal. However, this word did not appear in the query 2, hence leading to the lower similarity scores produced by SBERT model and lowering the final nDCG score.

Table 4: nDCG scores in the CRT subtask. The symbol † indicated that RUN_ID did not use the cosine similarity scores but only based on the estimated reading comprehension.

RUN_ID	SBERT Type	top-m	Keywords Type	nDCG
1†	Fast-Mini	4	T1	0.5856
0	Fast-Mini	4	T1	0.6929
2	Fast-Mini	5	T1	0.7178
3	Fast-Mini	6	T1	0.7245
4	Fast-Mini	7	T1	0.7215
5	Fast-Mini	8	T1	0.7215
6	Mini	6	T1	0.7153
8	Base	6	T1	0.7149
9	Fast-Mini	6	T2	0.7271
10	Fast-Mini	6	T3	**0.7295**
11	Fast-Mini	6	T4	0.7164

7 CONCLUSIONS

In this paper, we present our approach to address the RCIR task. Our team participated in both subtasks of the challenge including CET and CRT. Regarding the CET subtask, we decided to employ conventional machine learning models due to the limited number of samples in the dataset. We conducted an extensive experiment to select the appropriate models with fine-tuned hyperparameters for 3 scenarios, which were subject-independent, subject-dependent, and general. The results showed that the support vector machine

classification model obtained the best evaluation metrics in the test set across all scenarios at 0.5993. In terms of CRT subtask, we performed the retrieval by combining the result generated in the CET subtask with the similarity scores between the topics and the content of the reading paragraphs which was computed by utilised SBERT model. We achieved the nDCG at 0.7296 by using the small and compact model which was better than the original model having a larger structure.

For future work, we can fine-tune the ERNIE or find another language model that can produce a good paragraph embedding. The questions in each text that were used to ask the readers are currently discarded. It would be better if we integrate this information into the model. As mentioned in section 3, volunteers were required to read in 4 different behaviors. Although the labels of these reading conditions are not provided, they are expected to contain information that can help to distinguish the comprehension scores.

ACKNOWLEDGMENTS

This publication has emanated from research supported in party by research grants from ADAPT Centre, Insight Centre for Data Analytics and Centre for Research Training in Artificial Intelligence funded by Science Foundation Ireland Research Centres Programme under grant numbers SFI/12/RC/2289, SFI/13/RC/2106, 13/RC/2106_P2 and 18/CRT/6223. This work was also co-funded by the European Regional Development Fund.

REFERENCES

[1] Hervé Abdi and Lynne J Williams. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics* 2, 4 (2010), 433–459.

[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*. 1–5.

[3] Ralf Biedert, Jörn Hees, Andreas Dengel, and Georg Buscher. 2012. A robust realtime reading-skimming classifier. In *Proceedings of the Symposium on Eye Tracking Research and Applications*. 123–130.

[4] Jonathan FG Boisvert and Neil DB Bruce. 2016. Predicting task from eye movements: On the importance of spatial distribution, dynamics, and image features. *Neurocomputing* 207 (2016), 653–668.

[5] Vladimir Cherkassky and Yunqian Ma. 2004. Practical selection of SVM parameters and noise estimation for SVM regression. *Neural networks* 17, 1 (2004), 113–126.

[6] 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.

[7] John M Henderson, Svetlana V Shinkareva, Jing Wang, Steven G Luke, and Jenn Olejarczyk. 2013. Predicting cognitive state from eye movements. *PloS one* 8, 5 (2013), e64937.

[8] Shoya Ishimaru, Kensuke Hoshika, Kai Kunze, Koichi Kise, and Andreas Dengel. 2017. Towards reading trackers in the wild: Detecting reading activities by EOG glasses and deep neural networks. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*. 704–711.

[9] Johanna K Kaakinen and Jukka Hyönä. 2007. Perspective effects in repeated reading: An eye movement study. *Memory & cognition* 35, 6 (2007), 1323–1336.

[10] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017).

[11] M Krzywinski, N Altman, et al. 2017. Points of Significance: Principal component analysis. *Nature Methods* 14 (2017), 641–642.

[12] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683* (2017).

[13] Jiayu Li, Ziyi Ye, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma. 2020. THUIR at the NTCIR-15 Micro-activity Retrieval Task. In *Proceedings of the NTCIR-15 Conference*.

[14] Silvia Makowski, Lena A Jäger, Ahmed Abdelwahab, Niels Landwehr, and Tobias Scheffer. 2018. A discriminative model for identifying readers and assessing text comprehension from eye movements. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 209–225.

[15] Keith Rayner, Kathryn H Chace, Timothy J Slattery, and Jane Ashby. 2006. Eye movements as reflections of comprehension processes in reading. *Scientific studies of reading* 10, 3 (2006), 241–255.

[16] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[17] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 8968–8975.

[18] Geoffrey Underwood, Alison Hubbard, and Howard Wilkinson. 1990. Eye fixations predict reading comprehension: The relationships between reading skill, reading speed, and visual inspection. *Language and speech* 33, 1 (1990), 69–81.

[19] Zhixiang Xu, Gao Huang, Kilian Q Weinberger, and Alice X Zheng. 2014. Gradient boosted feature selection. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 522–531.