KNUIR at the NTCIR-16 RCIR: Predicting Comprehension Level using Regression Models based on Eye-Tracking Metadata

Yumi Kim
Kyungpook National University
South Korea
yumikim@knu.ac.kr

Aluko Ademola
Kyungpook National University
South Korea
joshuaokupevi5000@gmail.com

Jeong Hyeun Ko
Kyungpook National University
South Korea
jhko@knu.ac.kr

Heesop Kim
Kyungpook National University
South Korea
heesop@knu.ac.kr

ABSTRACT

We participated in the CET sub-task of the NTCIR-16 RCIR. In order to participate in the NTCIR-16 reading comprehension information retrieval (RCIR) CET sub-task, we adopted five regression models: Linear Regression, Random Forest Regressor, Gradient Boosting Regressor, eXtreme Gradient Boosting (XGB) Regressor, and Voting Regressor. We submitted the prediction results of test data to NTCIR-16 and analyzed the obtained results.

Throughout the analysis, we found that Gradient Boosting and Random Forest Regressor generally show better performance with Spearman's ρ of 0.53 and 0.57, respectively. In addition, the feature importance analysis indicated that each participant shows different eye-tracking tendencies for their reading comprehension. Findings in our work may bring insight into the understanding of human reading and information seeking processes with the help of eye-tracking systems by applying various regression models.

KEYWORDS

Reading comprehension, Data prediction, Machine learning, Regression models, Information Retrieval, Eye-tracking

TEAM NAME

KNUIR

SUBTASKS

RCIR CET subtask

1 INTRODUCTION

Machine reading comprehension (MRC), particularly real-world reading comprehension (RC) is one of the most challenging tasks in information retrieval (IR) researches involving multiple tasks such as reading, processing, comprehending, inferencing, reasoning, and summarizing [1, 2]. In recent years, a number of deep learning models have been adopted to some simplified MRC task settings, whose performance were close to or even better than human beings. However, understanding of the behavior patterns using eye-tracking remain under investigated.

The purpose of our study is to compare the prediction performance among the five regression models, that is, Linear Regression, Random Forest Regressor, Gradient Boosting Regressor, eXtreme Gradient Boosting (XGB) Regressor, and Voting Regressor at NTCIR-16 reading comprehension information retrieval (RCIR) CET sub-task [3]. Findings in our work may bring insight into the understanding of human reading and information seeking processes, and help the machine to better meet users' information needs.

The rest of this paper is organized as follows. Related works are in Section 2, followed by our research methods in Section 3. Section 4 describes the settings of experiments, and finally the conclusions and future works suggest in Section 5.

2 RELATED WORK

2.1 Reading Comprehension with Eye-traking approaches in IR.

Nishida et al [4] developed a Retrieve-and-Read model based on the bi-directional attention flow (BiDAF) model [5] for supervised multi-task learning (MTL) of IR and RC tasks that shares its hidden layers between the two tasks and minimizes the joint loss. This model with a telescoping setting exceeded the state-of-the-art by a significant margin on a machine reading at scale (MRS) task, answering Stanford Question Answering Dataset (SQuAD) questions using the full Wikipeida.

Zheng et al [2] investigated human's reading behavior patterns during RC tasks, where 32 users were recruited to complete 60 distinct tasks. By analyzing the collected eye-tracking data and answers from participants, they proposed a two-stage reading behavior model, in which the first stage is to search for possible answer candidates and the second stage is to generate the final answer through a comparison and verification process. They also found that human's attention distribution is affected by both question-dependent factors, e.g., answer and soft matching signal with questions, and question-independent factors, e.g., position, inverse document frequency (IDF) and Part-of-Speech tags of words. They extracted features derived from the two-stage reading behavior model to predict human's attention signals during reading comprehension, which significantly improved performance in the MRC task.

Liu and Mao [6] conducted an eye-tracking study to investigate how human assessors read a document during relevance judgement tasks and adopted the findings in building a novel retrieval model that can better approximate human's relevance judgment. They also conducted another eye-tracking study to investigate human's reading behavior when completing the reading comprehension task. They built a prediction model for user attention and leverage the predicted attention signals to improve the machine reading comprehension model.

2.2 Machine Learning models in IR

Mewada et al [7] proposed a model based on shape extraction and room identification of the building's floor plan using Linear Regression model for automatic room information retrieval. The proposed model is tested on the Computer Vision Center-Floor Plan (CVC-FP) dataset with an average room detection accuracy of 85.71% and room recognition accuracy of 88%.

Breiman [8] argued that a Random Forests model showed an exceptional prediction accuracy, and this accuracy is attained for a wide range of settings for the single tuning parameter. Segal [9] revisited the formulation of Random Forests and investigate prediction performance on real-world and simulated datasets for which maximally sized trees do overfit. These explorations reveal that gains can be realized by additional tuning to regulate tree size via limiting the number of splits and/or the size of nodes for which splitting is allowed. Nonetheless, even in these settings, good performance for Random Forests can be attained by using larger than default primary tuning parameter values.

Natekin and Knoll [10] gave a tutorial into the methodology of Gradient Boosting methods with a strong focus on machine learning aspects of modeling, and discussed the handing of model complexity. They also presented three practical examples of Gradient Boosting applications and analyzed them comprehensively.

Palotti et al [11] found that machine learning techniques were more suitable to estimate health web page understandability than traditional readability formulae. They used the XGB Regressor as well as the Simple Measure of Gobbledygook (SMOG). In XGB, for assessed documents they used 10-fold cross validation, training XGB on 90% of the data, and used its predictions for the remaining 10%. For unassessed documents, they trained XGB on all assessed data and applied this model to generate predictions.

Kades et al [12] studied to optimally leverage BERT for the task of assessing the semantic textual similarity of clinical text data. They adapted three approaches: Voting Regression, M-Heads, and Medication Graph. The Voting Regression approach showed an improvement of the Pearson correlation coefficient of the training set, however, for the test set, the performance decreased. They observed that the success of the different methods strongly depended on the underlying dataset.

3 METHODS

3.1 Implementation

Python and Scikit-learn packages are mainly used for implementation of our CET prediction model. Python is a simple, powerful, and interpreted programming language and it has an active and supportive community of the developers [13]. It provides flexibility, extensibility, and interactivity, as well [14]. Scikit-learn is an open source python module integrating a wide range of machine learning algorithms [15–17] and it has also simple and efficient libraries for data prediction and analysis [18]. Scikit-learn is one of the most commonly used python libraries for machine learning. Our CET prediction model is mainly implemented using scikit-learn, Pandas and NumPy. Pandas is a python library of rich data structures and tools for analysis and manipulation in statistics, finance, social sciences, and many other fields [19]. NumPy is a python library for scientific computing [20].

3.2 Dataset

The dataset obtained in the NTCIR-16 RCIR Task is structured into 9 directories (from 0000 to 0008). Each directory contains one volunteer's reading data and other associating metadata (for model training) [3]. The reading data consists of comprehension score, topic and text ids, the duration of reading, and the number of total words. The comprehension score, denoted as *c_score*, is defined by the number of correct answers by a volunteer to three multiple choice questions per paragraph ($c\_score \in \{0, 1, 2, 3\}$). Thus, in this paper, we predict the comprehension score of each volunteer.

Other associated metadata includes the pre-computed 302 features from the volunteer's eye-tracking data. While participants were reading the text, their eye movements were measured with an eye-tracker system. An eye-tracker is a device that tracks the position of the saccade by detecting movement of the pupil. The eye-tracker system can measure what people look at over time [21].

3.3 Comparison of machine learning regression algorithms

In this work, we adopted five regression models in order to predict target results. Among the various regression models, we select five regression models: *i*) Linear Regression, *ii*) Random Forest Regressor, *iii*) Gradient Boosting Regressor, *iv*) XGB Regressor, and *v*) Voting Regressor.

Ensemble methods involve combining multiple diverse machine learning models with the aim of improving the prediction performance [22]. We adopted two representative ensemble models: Random Forest and Gradient Boosting, which utilize the technique of bagging and boosting, respectively. Random Forest Regressor model uses the average of multiple decision trees for various subsamples of the dataset to increase prediction accuracy and control overfiting. The Gradient Boosting Regressor model, can handle non-linear correlation between input data and target result as well as correlation between features [23].

We also used the XGB Regressor, which is generally known for providing highly accurate results among regression models. In addition, we applied the Voting Regressor which combines the performances of the other four models to make predictions. In our analysis, we compared the importance of features derived from regression models, as implemented in XGBoost Python library [24].

4 EXPERIMENTS

Training data is automatically shuffled and divided into training data and validation data at a ratio of 7:3. We utilized the validation data to evaluate the performance of each model, and compared the performance of each model with Spearman's rank correlation

Table 1: Training parameters of each model.

Gradient Boost Regressor	Random Forest Regressor	Linear Regression	Voting Regressor	XGB Regressor
loss=’squared_error’, *learning_rate*=0.1, *n_estimators*=100, *subsample*=1.0, *criterion*=’friedman_mse’, *max_depth*=3, *init*=None, *random_state*=1, *max_features*=None, others set as default	*n_estimators*=100, *criterion*=’squared_error’, *max_depth*=None, *random_state*=1, *verbose*=0, others set as default	*fit_intercept*=True, *normalize*=’deprecated’, *copy_X*=True, *n_jobs*=None, *positive*=False	*weights*=None, *n_jobs*=None, *verbose*=False	*n_estimators*=100, *learning_rate*=0.08, *gamma*=0, *subsample*=0.75, *colsample_bytree*=1, and *max_depth*=7, others set as default

Table 2: Spearman’s rank correlation coefficient (Spearman’s ρ) and p-value of each model.

	Spearman’s ρ (p-value) of Volunteer								
	$1st$	$2nd$	$3rd$	$4th$	$5th$	$6th$	$7th$	$8th$	$9th$
Gradient Boost Regressor	0.71 (< 0.0002)	0.63 (< 0.0018)	0.27 (< 0.2330)	0.59 (< 0.0036)	0.22 (< 0.3310)	0.47 (< 0.0264)	0.15 (< 0.5144)	0.38 (< 0.0839)	0.86 (< 2.1227)
Random Forest Regressor	0.70 (< 0.0002)	0.76 (< 4.680)	0.22 (< 0.0334)	0.61 (< 0.0023)	0.12 (< 0.6082)	0.36 (< 0.0967)	0.06 (< 0.7980)	0.54 (< 0.0096)	0.77 (< 3.2613)
Linear Regression	0.27 (< 0.2079)	0.31 (< 0.1537)	0.27 (< 0.2188)	0.12 (< 0.5941)	0.06 (< 0.7854)	0.46 (< 0.0305)	0.07 (< 0.7681)	0.20 (< 0.3720)	0.41 (< 0.0588)
XGB Regressor	0.69 (< 0.0004)	0.64 (< 0.0012)	0.35 (< 0.1087)	0.59 (< 0.0041)	0.11 (< 0.6400)	0.46 (< 0.0324)	0.04 (< 0.8585)	0.47 (< 0.0276)	0.80 (< 8.4717)
Voting Regressor	0.71 (< 0.0001)	0.70 (< 0.0003)	0.35 (< 0.1114)	0.36 (< 0.1004)	0.11 (< 0.6123)	0.62 (< 0.0020)	0.11 (< 0.6354)	0.37 (< 0.0926)	0.81 (< 4.2279)

Table 3: Running Result from NTCIR-16 RCIR *

	ρ	p-value
Gradient Boost Regressor	0.53186	< 0.00000
Random Forest Regressor	0.57061	< 0.00000
Linear Regression	0.05021	< 0.46292
Voting Regressor	0.31124	< 0.00000

* The result of XGB Regressor is not included due to late submission.

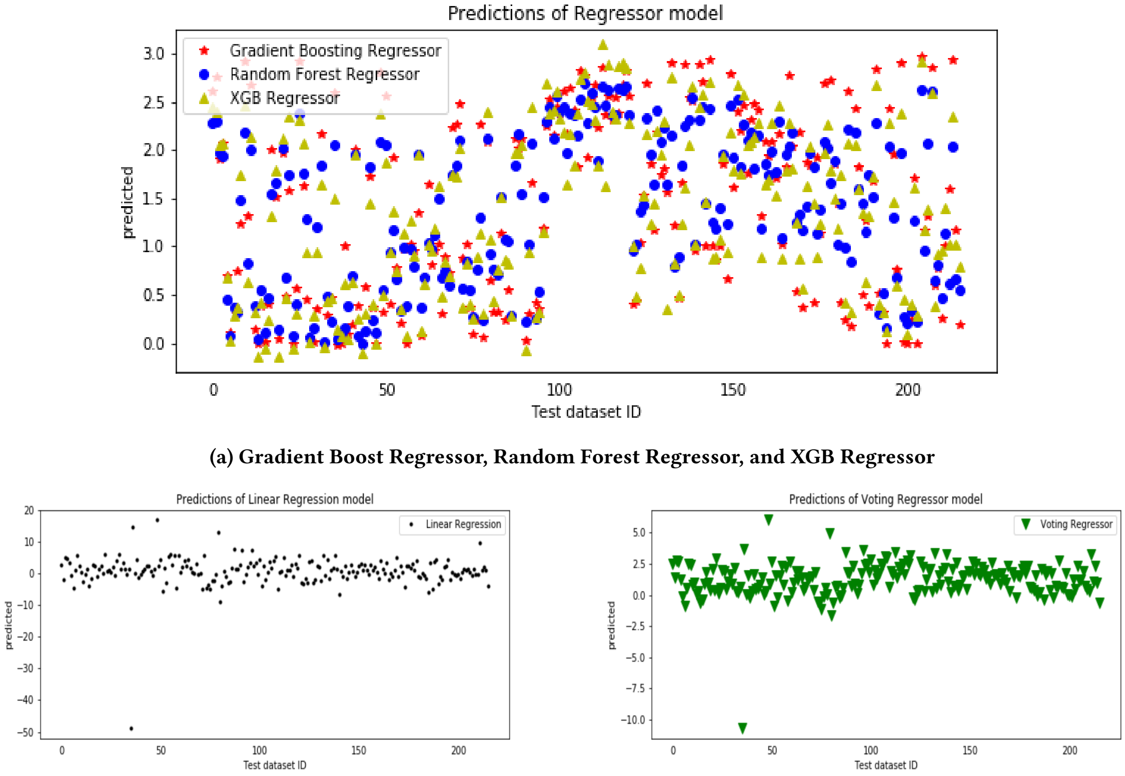
coefficient (Spearman’s ρ) and prediction values. Five models were trained using the parameters in Table 1.

Table 2 shows each volunteer’s Spearman’s ρ and p-value for each regression model of on the validation data. As shown in Table 2, four regression models except Linear Regression show similar Spearman’s ρ. On the contrary, Linear Regression model provides relatively poor performance in our validation data. Furthermore, each volunteer shows different performance. For example, 1st, 2nd, 4th, and 9th volunteers are generally predicted well by the regression models.

We submitted the prediction results of test data to NTCIR-16 and obtained the evaluation results. Figure 1 shows the submitted prediction results. As shown in Figure 1, Gradient Boosting Regressor, Random Forest Regressor, and XGB Regressor perform better with the prediction values reside from 0 to 3. On the contrary, Linear Regression and Voting Regressor perform worse with some predicted values that are out of the range. Table 3 shows the actual results of the submitted data from NTCIR-16. As we expected, Table 3 shows that Gradient Boosting and Random Forest Regressor show better performance.

We also analyzed important features of two models with the best performance, Gradient Boosting and Random Forest Regressor. *Feature importance* is a measure of how important a feature is in predicting the dependent variable. Figure 2 shows top 15 feature importances of four volunteers (1st, 2nd, 4th, and 9th) with high performance in the two models.

As shown in Figure 2, only a few features affects the performance among the 302 features. In addition, the most important features of volunteers are different, which implies that each participant shows different eye-tracking tendencies for their reading comprehension. However, RATE_X_BWD, RATE_BLINK, FIXA_X_FWD_tr_max, and FIXA_X_FWD_max-min are shared common features used to predict comprehension level. Details of these features are as follows:



(a) Gradient Boost Regressor, Random Forest Regressor, and XGB Regressor

(b) Linear Regression

(c) Voting Regressor

Figure 1: Scatter plots for prediction with test data.

- RATE_X_BWD: feature related to frequency of the eyes moving back and fixating on certain points,
- RATE_BLINK: number of blinks divided by the total words in the text, and
- FIXA_X_FWD_tr_max, FIXA_X_FWD_max-min: features related to forward and backward movement distances.

5 CONCLUSIONS

5.1 Concluding Remark

In this paper, we implemented and compared five regression models in order to predict the comprehension score based on reading data and eye-tracking metadata. We evaluated the performance of each model with Spearman's rank correlation coefficient (Spearman's ρ) and prediction values. Based on our findings, we showed that Gradient Boosting and Random Forest Regressor show better performance. The Spearman's ρ values of the two models benchmarked on the NTCIR-16 RCIR test set are 0.53 and 0.57, respectively.

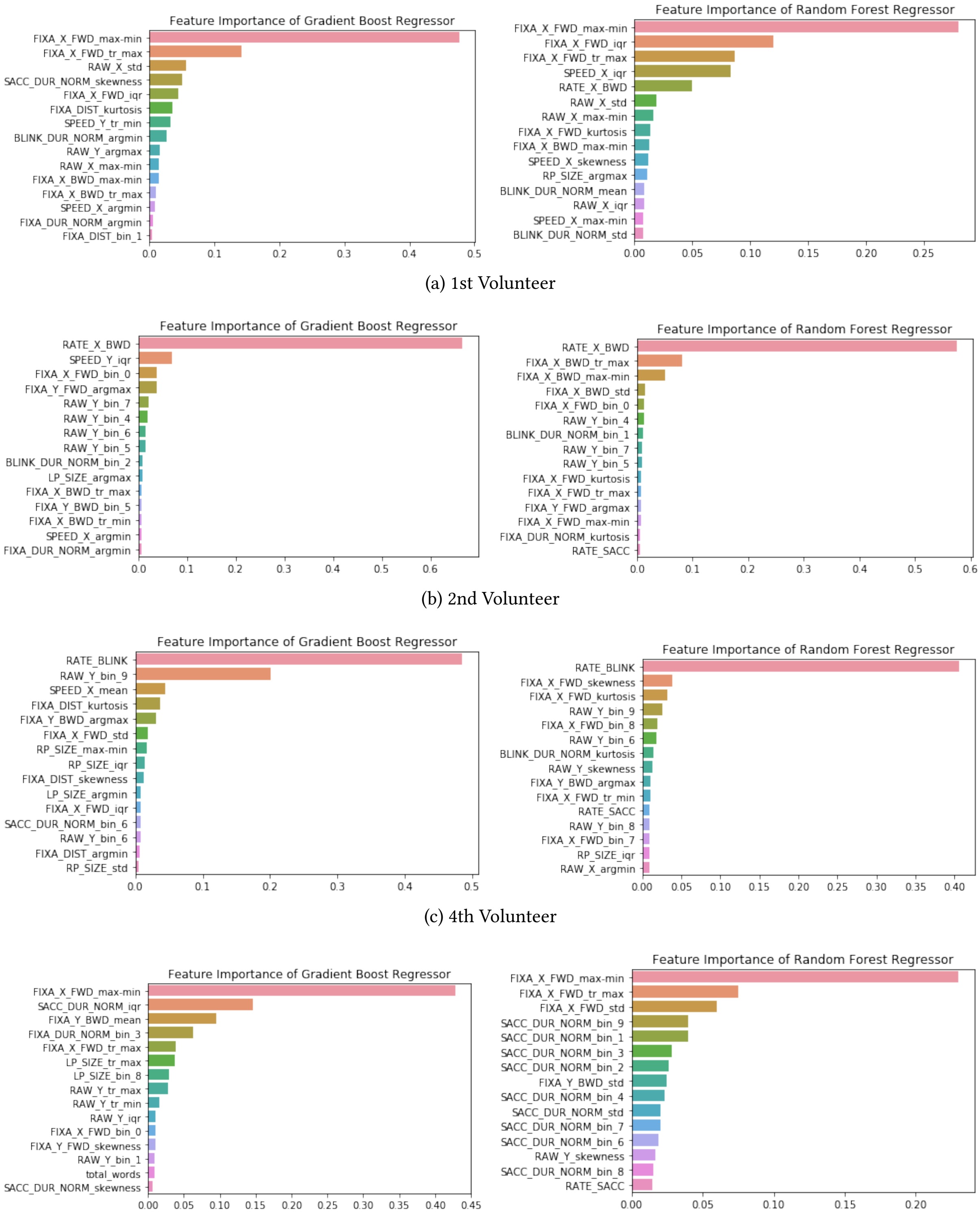
The feature importance analysis indicated that only a few features affects the performance among the 302 features. In addition, the most important features of volunteers are varied, which implies that each participant shows different eye-tracking tendencies for their reading comprehension. We found that RATE_X_BWD, RATE_BLINK, FIXA_X_FWD_tr_max, and FIXA_X_FWD_max-min affect the prediction result highly as important features.

5.2 Future Work

One of most important factors for deriving a learning model with a high performance is known as input data from features selection [25]. Feature selection methods [26, 27] not only decrease complexity of the model and mitigate overfitting problems, but also select the most important features from the dataset [27]. In other words, a much more meaningful model can be derived with optimized features.

In our experiments, we took all the features of the eye-tracking metadata in training data. As shown in Figure 2, no matter how many features we create and apply, not all of them are important. A large number of features not only requires a lot of computation time, cost and human efforts, but also leads to complex models and poor performance. Thus, our team is still working on feature selection for better prediction models. For example, we are filtering the features of dataset by using the low variance features, high correlation features, and univariate feature selection.

We are also planning to analyze the relationship between the feature importance and the prediction results in more detail. Throughout the further analysis, we will conduct more study on the analysis of the relationship between personalized comprehension level and eye-tracking metadata, too.



(a) 1st Volunteer

(b) 2nd Volunteer

(c) 4th Volunteer

(d) 9th Volunteer

Figure 2: Feature Importance of $1st$,$2nd$,$4th$, and $9th$ Volunteer

REFERENCES

[1] P Bajaj, D Campos, N Craswell, L Deng, J Gao, X Liu, R Majumder, A McNamara, B Mitra, T Nguyen, et al. 2016. A human generated MAchine Reading COmprehension dataset. *arXiv preprint arXiv:1611.09268* (2016).

[2] Yukun Zheng, Jiaxin Mao, Yiqun Liu, Zixin Ye, Min Zhang, and Shaoping Ma. 2019. Human behavior inspired machine reading comprehension. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 425–434.

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

[4] Kyosuke Nishida, Itsumi Saito, Atsushi Otsuka, Hisako Asano, and Junji Tomita. 2018. Retrieve-and-read: Multi-task learning of information retrieval and reading comprehension. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 647–656.

[5] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603* (2016).

[6] Yiqun Liu and Jiaxin Mao. 2020. " Revisiting information retrieval tasks with user behavior models" by Yiqun Liu and Jiaxin Mao with Martin Vesely as coordinator. *ACM SIGWEB Newsletter* Autumn (2020), 1–8.

[7] Hiren K Mewada, Amit V Patel, Jitendra Chaudhari, Keyur Mahant, and Alpesh Vala. 2020. Automatic room information retrieval and classification from floor plan using linear regression model. *International Journal on Document Analysis and Recognition (IJDAR)* 23, 4 (2020), 253–266.

[8] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[9] Mark R Segal. 2004. Machine learning benchmarks and random forest regression. (2004).

[10] Alexey Natekin and Alois Knoll. 2013. Gradient boosting machines, a tutorial. *Frontiers in neurorobotics* 7 (2013), 21.

[11] Joao Palotti, Guido Zuccon, Allan Hanbury, . 2019. Consumer health search on the web: study of web page understandability and its integration in ranking algorithms. *Journal of medical Internet research* 21, 1 (2019), e10986.

[12] Klaus Kades, Jan Sellner, Gregor Koehler, Peter M Full, TY Emmy Lai, Jens Kleesiek, and Klaus H Maier-Hein. 2021. Adapting Bidirectional Encoder Representations from Transformers (BERT) to Assess Clinical Semantic Textual Similarity: Algorithm Development and Validation Study. *JMIR medical informatics* 9, 2 (2021), e22795.

[13] Guido Rossum. 1995. Python reference manual. (1995).

[14] Michel F Sanner . 1999. Python: a programming language for software integration and development. *J Mol Graph Model* 17, 1 (1999), 57–61.

[15] JC Gertrudes, Vinícius Gonçalves Maltarollo, RA Silva, Patricia Rufino Oliveira, Kathia Maria Honorio, and ABF Da Silva. 2012. Machine learning techniques and drug design. *Current medicinal chemistry* 19, 25 (2012), 4289–4297.

[16] Angelica Nakagawa Lima, Eric Allison Philot, Gustavo Henrique Goulart Trossini, Luis Paulo Barbour Scott, Vinícius Gonçalves Maltarollo, and Kathia Maria Honorio. 2016. Use of machine learning approaches for novel drug discovery. *Expert opinion on drug discovery* 11, 3 (2016), 225–239.

[17] Lu Zhang, Jianjun Tan, Dan Han, and Hao Zhu. 2017. From machine learning to deep learning: progress in machine intelligence for rational drug discovery. *Drug discovery today* 22, 11 (2017), 1680–1685.

[18] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, . 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12, Oct (2011), 2825–2830.

[19] Wes McKinney . 2011. Pandas: a foundational Python library for data analysis and statistics. *Python for High Performance and Scientific Computing* 14, 9 (2011).

[20] Travis E Oliphant. 2006. *A guide to NumPy*. Vol. 1. Trelgol Publishing USA, Massachusetts,USA.

[21] Gho Kim. 2017. Visual Understanding of Advertising Through Eye-tracking Methodology. *The Korean Journal of Advertising and Public Relations* 19, 2 (2017), 41–84.

[22] Omer Sagi and Lior Rokach. 2018. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, 4 (2018), e1249.

[23] Chen Zheng, Clara Grzegorz Kasprowicz, and Carol Saunders. 2017. Customized Routing Optimization Based on Gradient Boost Regressor Model. *arXiv preprint arXiv:1710.11118* (2017).

[24] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.

[25] Ulrike Grömping. 2009. Variable importance assessment in regression: linear regression versus random forest. *The American Statistician* 63, 4 (2009), 308–319.

[26] Martin Eklund, Ulf Norinder, Scott Boyer, and Lars Carlsson. 2014. Choosing feature selection and learning algorithms in QSAR. *Journal of Chemical Information and Modeling* 54, 3 (2014), 837–843.

[27] Mohammad Goodarzi, Bieke Dejaegher, and Yvan Vander Heyden. 2012. Feature selection methods in QSAR studies. *Journal of AOAC International* 95, 3 (2012), 636–651.