NTCIR-18: Proceedings of the 18th NTCIR Conference on Evaluation ofInformation Access Technologies, June 10-13, 2025, Tokyo, Japan
DOI: https://doi.org/10.20736/0002002026

UCLWI at the NTCIR-18 AEOLLM Task: A Low-Cost Comparison of RAGs

Xiao Fu
University College London
London, UK
xiao.fu.20@ucl.ac.uk

Navdeep Singh Bedi
Università della Svizzera italiana
Lugano, Switzerland
navdeep.singh.bedi@usi.ch

Noriko Kando
National Institute of Informatics
Tokyo, Japan
Noriko.Kando@nii.ac.jp

Fabio Crestani
Università della Svizzera italiana
Lugano, Switzerland
fabio.crestani@usi.ch

Aldo Lipani
University College London
London, UK
aldo.lipani@ucl.ac.uk

Abstract

The UCLWI team participated in the Automatic Evaluation of LLMs (AEOLLM) task of the NTCIR-18 [2]. We propose an efficient evaluation pipeline for Retrieval-Augmented Generation (RAG) systems tailored for low-resource settings. Our method uses ensemble similarity measures combined with a logistic regression classifier to assess answer quality from multiple system outputs using only the available queries and replies. Experiments across diverse tasks demonstrate competitive accuracy and reasonable correlation with ground truth rankings, establishing our approach as a reliable metric.

CCS Concepts

• **Information systems** → **Retrieval effectiveness**; **Question answering**; *Language models.*

Keywords

Information Retrieval, Evaluation, RAG system, Ensemble

Team Name

UCLWI

Subtasks

AEOLLM

1 Introduction

Retrieval-augmented generation (RAG) synergizes large language models (LLMs) with dedicated retrieval mechanisms to generate text that is both contextually relevant and factually grounded [6]. This paradigm has shown notable versatility in applications such as open-domain question answering and conversational agents. However, the evaluation of RAG systems remains a significant challenge. Traditional text generation metrics often fail to capture the nuances of factual accuracy and retrieval quality, while manual annotation methods can be prohibitively time-consuming and subjective.

Recent research has introduced evaluation strategies tailored specifically for RAG systems. These approaches typically combine automatic metrics, such as BLEU, ROUGE, and BERTScore, with retrieval-specific measures (e.g., precision, recall, and F1 score) or leverage LLM-based evaluations [1, 7, 8, 10]. Although these methods have advanced the field, their reliance on extensive computational resources and large annotated corpora limits their applicability in resource-constrained settings.

In this study, we address the challenge of RAG evaluation in low-resource environments, where neither additional corpora nor GPUs are available. Our pipeline, developed within the NTCIR AEOLLM framework, processes single queries answered by multiple RAG systems [2]. By leveraging an ensemble strategy that analyzes the similarity between generated answers and their corresponding queries, our method not only achieves the highest agreement but also attains the second-best accuracy relative to the ground truth. These results underscore its potential as an efficient and effective evaluation strategy for RAG systems.

2 Background

2.1 Retrieval-Augmented Generation Systems

Retrieval-augmented generation (RAG) systems integrate large language models with external retrieval components to produce outputs that are both contextually coherent and factually grounded. Lewis et al. [6] introduced a RAG framework that augments language models with a retrieval mechanism to tackle knowledge-intensive tasks. Building upon this paradigm, Guu et al. [4] proposed the REALM approach, which incorporates retrieval during pre-training and Izacard and Grave [5] further demonstrated the benefits of fusing retrieval with generative models in open-domain question answering.

2.2 Evaluation of RAG Systems

Evaluating RAG systems poses unique challenges as it requires assessing both the quality of the generated text and the effectiveness of the retrieval process. Standard automatic metrics, including BLEU [8], ROUGE [7], and BERTScore [10], have been widely used to evaluate text generation quality. However, these metrics often fall short in measuring retrieval performance and factual correctness. To overcome these limitations, recent work has explored the use of retrieval-specific metrics and even leveraged LLM-based evaluations [1] to provide a more comprehensive assessment.

2.3 Ensemble Methods in System Evaluation

Ensemble methods have been effectively used to enhance the reliability and performance of various machine learning systems by

Algorithm 1 RAG Evaluation Score Prediction

Require:
- A set of queries $Q = \{q_1, q_2, \ldots, q_m\}$.
- For each query q, a set of n replies $R = \{r_1, r_2, \ldots, r_n\}$.
- A similarity function $\text{sim}(\cdot, \cdot)$ to compute similarity between texts.
- A classifier function $\text{Classifier}(\cdot)$ that predicts a score s from input features.

Ensure: A set S containing tuples (q, r, s) for each query-reply pair.

$S \leftarrow \emptyset$
for each query $q \in Q$ **do**
  for each reply $r \in R$ **do**
    $F \leftarrow \emptyset$ ▷ Initialize feature vector for reply r
    for each reply $r_j \in R$ **do**
      $f_j \leftarrow \text{sim}(r, r_j)$
      $F \leftarrow F \cup \{f_j\}$
    end for
    $f_q \leftarrow \text{sim}(q, r)$
    $F \leftarrow F \cup \{f_q\}$
    $s \leftarrow \text{Classifier}(F)$
    $S \leftarrow S \cup \{(q, r, s)\}$
  end for
end for
  return S

combining multiple models to offset individual weaknesses. In the context of RAG evaluation, ensemble strategies offer a promising avenue to integrate diverse evaluation signals, thereby yielding more robust quality predictions. Dietterich [3] provides a foundational overview of ensemble techniques, illustrating how aggregating multiple models can lead to significant performance gains—a concept that has influenced this work.

3 Design and Rationale

As described in Sections 1 and 2, this study evaluates replies from multiple RAG systems addressing the same query under low computational cost and without additional corpus access. We adopt an ensemble approach based on the assumption that the quality of an RAG's reply can be gauged by its similarity to other replies.

As shown in Fig. 1, the AEOLLM setting provides only the queries and the corresponding replies from each RAG system. We compute similarity scores between each reply and the original query and then use these scores as features in a classifier to predict the final quality of each reply.

Algorithm 1 details the process of our proposed model. In the next section, we provide further details on its implementation.

4 Details of implementation

4.1 Model Components

In this section, we detail the two main components of our pipeline: the similarity model and the classifier.

4.2 Similarity Model

To minimize computational cost while effectively calculating similarities, we utilize *static-similarity-mrl-multilingual-v1*[1] from *sentence-transformers* [9]. Unlike Transformer-based models (e.g., *all-MiniLM-L6-v2*[2]), this static model encodes text chunks into 1024-dimensional vectors, offering a balance between performance and computational efficiency. In our study, we compute similarities by obtaining embeddings for text chunks and measuring the cosine similarity between them.

4.3 Classifier

Logistic Regression Classification Implementation. The classifier employs a multinomial logistic regression model to predict answer quality on a 5-point scale (1–5). For each prediction, the input feature vector is 8-dimensional, comprising seven components representing the similarity scores between the candidate reply and each of the other replies ($f_1, \ldots, f_7$), along with one component capturing the similarity between the reply and the query (f_q). The feature matrix $F \in \mathbb{R}^{n \times 8}$ is standardized to have zero mean and unit variance.

The model is configured for multi-class classification using the multinomial scheme with the 'lbfgs' optimizer, running for a maximum of 1000 iterations. Given n samples, the model learns weight matrices $W \in \mathbb{R}^{5 \times 8}$ and bias terms $b \in \mathbb{R}^5$, with each row corresponding to a score class. The probability of assigning class k to an input x is computed as:

$$P(y = k \mid x) = \frac{\exp(W_k x + b_k)}{\sum_{j=1}^{5} \exp(W_j x + b_j)}.$$

The model parameters are optimized by maximizing the multinomial log-likelihood, and the final prediction is determined by:

$$\hat{y} = \arg\max_k \; P(y = k \mid x).$$

The trained model assigns discrete scores (1–5) to new answers based on their similarity features. These scores are subsequently converted to dense ranks, with equal scores receiving the same rank and subsequent ranks assigned consecutively, ensuring a consistent ordering of answer quality within each question group.

5 Results

Task	Accuracy	Kendall's Tau	Spearman
Dialogue Generation	0.7044	0.4913	0.5527
Text Expansion	0.5340	0.2758	0.3019
Summary Generation	0.7494	0.6114	0.6418
Non-Factoid QA	0.6905	0.4090	0.4311
Overall	0.6696	0.4468	0.4819

Table 1: Performance of our pipeline on the test set.

Our proposed pipeline performed comparable on both the test and final sets. On the test set, we obtained an overall accuracy of 0.6696, with agreement levels (above 0.4) observed in three out

[1] https://huggingface.co/sentence-transformers/static-similarity-mrl-multilingual-v1

[2] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

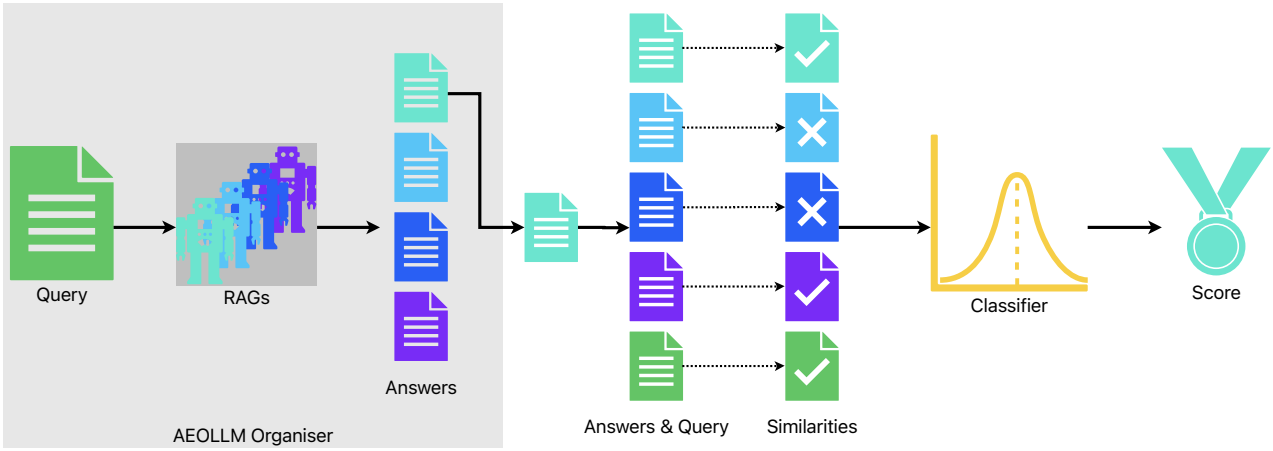

Figure 1: Overall structure of this work.

Task	Accuracy	Kendall's Tau	Spearman
Dialogue Generation	0.7756	0.5798	0.6426
Text Expansion	0.5266	0.3482	0.3815
Summary Generation	0.7273	0.5432	0.5763
Non-Factoid QA	0.6853	0.4105	0.4291
Overall	0.6787	0.4704	0.5074

Table 2: Performance of our pipeline on the final set.

of four tasks. Similarly, on the final set, we achieved an overall accuracy of 0.6787, with agreement levels similar to those on the test set.

At the task level, performance was consistent across all four tasks. In particular, Dialogue Generation and Summary Generation yielded relatively higher performance, followed by Non-Factoid QA, while Text Expansion consistently demonstrated the lowest performance.

At the time of writing, our approach attained the second-best overall accuracy and the highest Kendall's Tau and Spearman correlation scores on the final AEOLLM set.

In the next section, we analyze the performance and underlying models in further detail.

6 Analysis and Conclusion

Task	Avg Q Length	Avg R Length
Dialogue Generation	506.22	84.47
Text Expansion	131.5	1793.17
Summary Generation	1091.98	407.43
Non-Factoid QA	51.93	1142.0

Table 3: Average Query and Reply Lengths by Task

One notable observation from Section 5 is the significant variance in performance across tasks, which appears to stem from inherent differences in each task's characteristics. Table 3 presents the average lengths of queries and replies for each task, revealing marked disparities. When comparing Table 3 with Table 2, a clear trend emerges: tasks with longer queries and shorter replies tend to yield higher accuracy and better agreement in our pipeline.

This trend can be interpreted from two perspectives. On one hand, generation-based systems benefit from richer contextual information provided by longer queries, often leading to more accurate responses. On the other hand, the nature of the task itself plays a significant role. For instance, Summary Generation tasks require condensing text into concise, less diverse outputs, whereas Text Expansion tasks—where systems generate narratives from a given theme—tend to produce more varied responses. Consequently, the similarities among summaries are generally higher than those among expanded texts. This observation underscores a limitation of our pipeline: it relies on surface-level similarity metrics rather than a deeper semantic understanding to rank replies.

6.1 Weights in Classifier

Since all similarity scores have been standardized, the weights learned by the classifier on the training set can be directly interpreted as reflecting the relationship between these similarity measures and the corresponding quality scores. This makes it meaningful to analyze the classifier weights. Table 4 presents these weights, where each row includes 8 weights and 1 bias. The first 7 weights correspond to reply-to-reply similarity scores, and the 8th weight corresponds to the query-to-reply similarity score. In the first 5 rows for each task (corresponding to scores 1 to 5), the largest weight values are highlighted in bold and the smallest values are underlined. Because scores 4 and 5 are considered indicative of higher quality, we sum their corresponding weights to form an additional (sixth) row for each task; in this combined row, the largest

Table 4: Logistic Regression Parameters with Highlighted Extreme Features

Task	Class	R-R 1	R-R 2	R-R 3	R-R 4	R-R 5	R-R 6	R-R 7	Q-R 8	Bias
0	1	-0.1068$^{(5)}$	0.0402$^{(2)}$	-0.0808$^{(3)}$	-0.0998$^{(4)}$	-0.4810$^{(6)}$	-0.8308$^{(8)}$	-0.6910$^{(7)}$	**0.2767**$^{(1)}$	-1.3944
0	2	-0.0276$^{(3)}$	-0.1692$^{(7)}$	-0.0295$^{(4)}$	0.0654$^{(2)}$	-0.3904$^{(6)}$	-0.4155$^{(5)}$	-0.5905$^{(8)}$	**0.1629**$^{(1)}$	-0.4121
0	3	0.2796$^{(2)}$	0.2115$^{(3)}$	-0.1400$^{(8)}$	**0.3743**$^{(1)}$	0.1336$^{(4)}$	-0.0601$^{(6)}$	0.2599$^{(5)}$	-0.0748$^{(7)}$	0.6842
0	4	0.3030$^{(3)}$	0.3505$^{(2)}$	0.2772$^{(4)}$	-0.2269$^{(7)}$	0.1491$^{(5)}$	**0.4298**$^{(1)}$	0.0788$^{(6)}$	-0.3810$^{(8)}$	1.0157
0	5	-0.4481$^{(8)}$	-0.4330$^{(7)}$	-0.0269$^{(5)}$	-0.1131$^{(6)}$	0.5887$^{(3)}$	0.8766$^{(2)}$	**0.9429**$^{(1)}$	0.0162$^{(4)}$	0.1066
0	4+5	11	9	9	13	8	3	7	12	
1	1	-0.0392$^{(4)}$	0.1869$^{(2)}$	**0.3192**$^{(1)}$	-0.3523$^{(6)}$	-0.5888$^{(8)}$	-0.1947$^{(5)}$	-0.4194$^{(7)}$	0.0949$^{(3)}$	-0.0831
1	2	-0.4750$^{(7)}$	**0.3982**$^{(1)}$	-0.1815$^{(5)}$	0.2952$^{(2)}$	0.0488$^{(4)}$	0.0572$^{(3)}$	-0.5739$^{(8)}$	-0.4675$^{(6)}$	0.7544
1	3	0.1024$^{(4)}$	0.1224$^{(2)}$	-0.1319$^{(6)}$	**0.2898**$^{(1)}$	-0.2969$^{(7)}$	0.0610$^{(5)}$	-0.4028$^{(8)}$	0.1205$^{(3)}$	1.2505
1	4	0.1586$^{(2)}$	-0.3755$^{(8)}$	-0.1548$^{(6)}$	**0.2395**$^{(1)}$	0.1249$^{(3)}$	-0.0113$^{(4)}$	-0.1395$^{(5)}$	-0.1735$^{(7)}$	0.8596
1	5	0.2532$^{(4)}$	-0.3320$^{(7)}$	0.1491$^{(5)}$	-0.4721$^{(8)}$	0.7119$^{(2)}$	0.0878$^{(6)}$	**1.5356**$^{(1)}$	0.4256$^{(3)}$	-2.7813
1	4+5	6	15	11	9	5	10	6	10	
2	1	-0.3199$^{(5)}$	-0.7197$^{(6)}$	-0.2809$^{(4)}$	-1.0161$^{(8)}$	-0.0895$^{(3)}$	**0.6461**$^{(1)}$	-0.8294$^{(7)}$	0.3223$^{(2)}$	-3.3305
2	2	0.1056$^{(5)}$	0.2139$^{(4)}$	-0.4705$^{(7)}$	0.5948$^{(2)}$	**0.8959**$^{(1)}$	0.5169$^{(3)}$	-0.3934$^{(6)}$	-0.6836$^{(8)}$	-0.6081
2	3	-0.9310$^{(8)}$	0.2913$^{(3)}$	0.1482$^{(4)}$	0.4319$^{(2)}$	0.1363$^{(5)}$	0.1352$^{(6)}$	**0.4732**$^{(1)}$	-0.0915$^{(7)}$	1.1871
2	4	0.4945$^{(3)}$	0.2438$^{(4)}$	-0.0024$^{(5)}$	-0.0347$^{(6)}$	-0.4248$^{(7)}$	-0.7874$^{(8)}$	0.5358$^{(2)}$	**0.6535**$^{(1)}$	1.8194
2	5	**0.6509**$^{(1)}$	-0.0292$^{(5)}$	0.6056$^{(2)}$	0.0241$^{(4)}$	-0.5178$^{(8)}$	-0.5108$^{(7)}$	0.2139$^{(3)}$	-0.2007$^{(6)}$	0.9321
2	4+5	4	9	7	10	15	15	5	7	
3	1	-0.6849$^{(8)}$	-0.2905$^{(6)}$	-0.5107$^{(7)}$	0.0474$^{(2)}$	**0.5587**$^{(1)}$	-0.0972$^{(3)}$	-0.2541$^{(5)}$	-0.2317$^{(4)}$	-3.0199
3	2	-0.5310$^{(7)}$	**0.4316**$^{(1)}$	-0.6630$^{(8)}$	0.1053$^{(3)}$	0.2557$^{(2)}$	0.0374$^{(4)}$	-0.3257$^{(5)}$	-0.5171$^{(6)}$	-1.8783
3	3	**0.3214**$^{(1)}$	-0.5570$^{(8)}$	0.2456$^{(4)}$	-0.3920$^{(7)}$	-0.1086$^{(5)}$	-0.3317$^{(6)}$	-0.0901$^{(3)}$	0.1840$^{(2)}$	0.7111
3	4	-0.0530$^{(7)}$	0.2372$^{(5)}$	**0.7890**$^{(1)}$	0.2925$^{(4)}$	-0.3305$^{(8)}$	0.4501$^{(2)}$	0.2963$^{(3)}$	0.2100$^{(6)}$	1.9498
3	5	**0.9475**$^{(1)}$	0.1787$^{(4)}$	0.1391$^{(5)}$	-0.0531$^{(6)}$	-0.3753$^{(8)}$	-0.0586$^{(7)}$	0.3737$^{(2)}$	0.3548$^{(3)}$	2.2374
3	4+5	8	9	6	10	16	9	5	9	

weight is also highlighted in bold. With classifiers trained separately for each task, the performance of individual RAG systems can be discussed in detail.

Co-occurrence of Extreme Weights: An interesting observation is that extreme weight values tend to co-occur. Across 28 RAG-task pairs, there are 40 extreme values (maximums and minimums), yet only 18 RAG-task pairs exhibit extreme weights in the individual scores (1 through 5). Of these, only 6 cases have a single extreme weight, while the remaining 12 cases show multiple extreme weights. This pattern aligns with the intuition that if a RAG system has a high likelihood of achieving a certain score, it correspondingly has a lower chance of attaining other scores. Therefore, if we define scores 4 and 5 as representing "better" performance, then the rank of their weights can serve as an indicator of a RAG system's propensity for higher quality outputs. For example, assuming the order of RAGs in both the training and final sets remains unchanged, our analysis suggests that for task 0 (Dialogue Generation), RAG systems 6, 7, and 5 would perform best. Similarly, for task 1 (Text Expansion), RAG systems 7, 5, and 1 are likely to excel; for task 2 (Summary Generation), RAG systems 1, 7, and 3 would be preferable; and for task 3 (Non-Factoid QA), RAG systems 7, 3, and 1 are anticipated to deliver superior performance.

Influence of Query-to-Reply Similarity: Another notable observation is the impact of query-to-reply similarity weights. In tasks with longer queries, the query-to-reply similarity component plays a more critical role, as evidenced by the more extreme weight values. This alignment between the weights of query-to-reply similarity and the performance trends observed in Section 5 suggests that richer query contexts significantly contribute to the overall effectiveness of our evaluation pipeline.

6.2 Conclusion and Future Work

In summary, our study presents a comprehensive evaluation of RAG systems through a novel pipeline that leverages similarity metrics and classifier-based scoring. The analysis of task-level performance, classifier weights, and the role of query-to-reply similarities provides valuable insights into the strengths and limitations of our approach. While the reliance on similarity metrics offers computational efficiency, it also highlights the need for incorporating deeper semantic understanding in future work. Moving forward, we plan to explore advanced models that integrate both similarity and semantic understanding to further enhance the evaluation of RAG systems.

References

[1] Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv* (2023).

[2] Junjie Chen, Haitao Li, Zhumin Chu, Yiqun Liu, and Qingyao Ai. 2025. Overview of the NTCIR-18 Automatic Evaluation of LLMs (AEOLLM) Task. *arXiv preprint arXiv:2503.13038* (2025).

[3] Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*. Springer, 1–15.

[4] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*. PMLR, 3929–3938.

[5] Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282* (2020).

[6] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[7] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[8] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.

[9] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://arxiv.org/abs/1908.10084

[10] Tianyi Zhang, Varsha Kishore, FelixF. Wu, KilianQ. Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating Text Generation with BERT. *Cornell University - arXiv,Cornell University - arXiv* (Apr 2019).