RSLTOT at the TREC 2023 ToT Track

Reo Yoshikoshi[1] and Tetsuya Sakai[2]

[1] Sakai Laboratory, Waseda University, Tokyo, Japan
yoshikoshi@akane.waseda.jp
[2] Waseda University, Tokyo, Japan
tetsuyasakai@acm.org

Abstract. In this study, we focused on the situation that a user can recall only the movie's synopsis, character features, etc., but not the movie's title. In our experiment, we introduced systems based on TF–IDF and BERT. The results showed that our TF–IDF vectorizer is better than our BERT model if they are used individually. In addition, as each system showed different tendencies in the results, we tried a hybrid model combining these two systems. The results showed that combining these models outperformed the two component models.

1 Introduction

Users sometimes struggle to recall titles, character names, and content details of various consumed media. This phenomenon is known as the tip of the tongue (TOT) phenomenon[3][9]. During a TOT state, users may partially recall similar words or meanings but not the exact information. Elsweiler et al.[6] found that users in this state tend to experience stronger frustration compared to other memory loss conditions.

To ease the frustration caused by such situations, platforms have emerged where users can ask questions and gather answers. For example, "I Remember This Movie" helps users remember forgotten movie details by allowing them to post partially remembered plot summaries and contextual information leading to the viewing experience.

Arguello et al.[2] noted that existing information retrieval systems face difficulties when dealing with queries that lack precise names or identifiers, resulting in underperformance. As a result, users often turn to forum queries without attempting independent searches. This study aims to address these search needs by exploring multiple approaches to build an information retrieval system that can efficiently resolve such frustrations, aiming to provide a more suitable method for users and ease their frustration.

2 Data

In our experiments, we primarily used the data distributed by TREC 2023 ToT Track[1], which consists of a Wikipedia corpus and a set of questions posted to "I Remember This Movie"[3].

[3] https://irememberthismovie.com/

The set of questions is a subset of MS–TOT [2], and includes questions which had been posted from 2013 to 2018. This set also contains analyzed information, such as answer movie and sentence annotations. Answer movie information can be retrieved by using IDs and URLs, which uniquely identify Wikipedia articles and IMDb pages. Sentences were annotated by Arguello et al.[2] and indicate the type of information contained in each sentence.

The Wikipedia corpus is a collection of 231,852 pages that directly or indirectly related to the “audiovisual works” category. In order to enable to retrieve information from Infobox as dictionary format, texts from corpus are removed Wikimarkup and preprocessed. Additionally, the corpus contains 190,370 articles associated with IMDb identifiers, allowing for easy retrieval of external resources.

The dev dataset was used to evaluate the systems.

3 Experiments

3.1 Text Preparation

Firstly, We extracted text from the “abstract”, “synopsis” and “plot” sections as they are representative of the abstract and story line. Articles that did not have these sections were considered as empty text in this method. Secondly, we combined these section texts by padding them with space characters to create the movie’s text. This text is referred to as the ‘preprocessed Wikipedia text’.

There are only 150 questions that can be used as the training dataset. We acknowledge that this is too small to train a machine learning effectively, even if we used corpus-contained IMDb information. Therefore, we attempted to retrieve additional information from IMDb. In this method, we obtained 1,067 pieces of external information from IMDb by utilizing IMDb information from the corpus and Python. From this, we were able to extract 509 outline texts and 890 review texts from them.

We refer to both of these external text that are linked to preprocessed Wikipedia text and the train dataset as ‘IMDb texts’.

3.2 Metrics

We used 150 questions from the dev dataset provided by the data group as test queries and created systems to output ranked lists of movies. The entire preprocessed Wikipedia text serves as the set of target movies. The TREC ToT Track employs multiple metrics to evaluate system performance. However, for the sake of simplicity in this study, we used the rank of the retrieved movie. For the sake of interpretability, we used the relative rank, which is normalized using a corpus size of 231,852. In these metrics, an absolute rank of 1,000 corresponds to a relative rank of 0.004, while a relative rank of 0.2 corresponds to an absolute rank of 46,370.

We determined the system performance by using the average relative ranks of the entire dev dataset.

3.3 TF–IDF Approach

After transforming the TF–IDF vectorizer for the preprocessed Wikipedia texts, we used it to obtained vector representations of both the preprocessed Wikipedia and question texts. In this approach, we utilized `TfidfVectorizer` from the `scikit-learn` library in Python and set the maximum document frequency (`max_df`) to 0.75 to exclude stop words and frequently occurring words. We then calculated the cosine similarity between the question texts and preprocessed Wikipedia text, and sorted the results in descending order.

The performance of this approach is shown in Table 1 and its distribution is shown in Fig. 1. The answer for question ID 786 was ranked first in the output, while the answer for question ID 861 was ranked at $217,161$ (equivalent of 0.9366 relative ranking). 80.0% of all dev dataset have a relative ranking of 0.2 or less.

Table 1: Summary of the evaluated performance by using TF–IDF

Type	Relative Rank	Question ID
Average	0.1131	–
Minimum	4.313×10^{-6}	786
First Quartile	4.225×10^{-4}	–
Median	2.011×10^{-2}	–
Third Quartile	0.1211	–
Maximum	0.9366	861

3.4 BERT Approach

Next, we conducted an experiment using the BERT model. We used `bert-base-uncased`[5], is available on Hugging Face[4]. The train dataset, preprocessed Wikipedia text, and IMDb text were fed into the BERT model and it was retrained.

We thought that the contents of reviews on IMDb are close to the question text, while the outline text on IMDb is similar to preprocessed Wikipedia text. Therefore, the data was divided into two groups, as shown in Table 2: (A) and (B). Four combinations were used as training data for data augmentation.

For the machine learning policy, we used triplet loss[10] as the loss function. In this function, we set (A) to anchor, (B) to positive, and (B) from irrelevant movies to negative. This will lead to shorten the distance between anchor and positive in embedding space and keep away the negative. We also set $\alpha = 5.0$ as the margin in this loss function. However, the triplet loss of Wang et al.[10] has a weakness that learning may not progress after satisfying $d_p + \alpha \leq d_n$ where d_p is the anchor-positive distance and d_n is the anchor-negative distance.

Table 2: Combination of the used data

Question text(A)	overview / synopsis text(B)
train dataset reviews on IMDb	preprocessed Wikipedia text synopsis on IMDb

[4] https://huggingface.co/bert-base-uncased

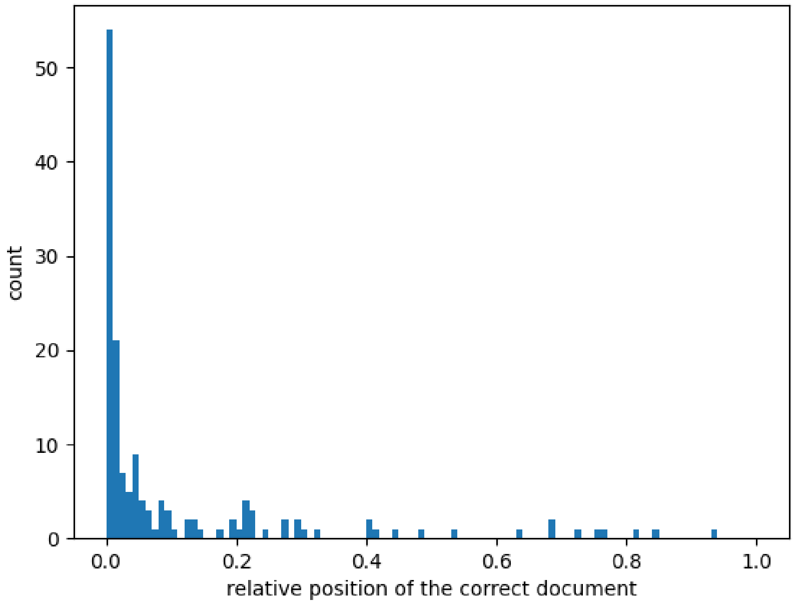

Fig. 1: Distribution of the relative rank of correct movies in outputs by using TF–IDF.

Therefore, we adopted the improved triplet loss introduced by Cheng et al.[4], and set beta to 1.0. This loss function is the method that adds $[d_p - \beta]_+$ to previous loss.[5] This will shorten d_p up to β and prevent the length of the vector from vanishing in the embedding space.

Deep metric learning[8] such as this learning method is good at learning under the condition; existence of unknown class, or too many classes, or too little training data. Therefore, this learning method seems to be suitable for this task. In addition, this triple loss is the loss function for image retrieval tasks, but according to Hoffer et al.[7], this is useful for wide range tasks that require metric learning. Therefore, we expected this approach to perform well even in the text-based task.

We calculated the Euclidean distance between the question text and preprocessed Wikipedia text, and sorted them in ascending order.

The performance of this approach is shown in Table 3 and its distribution is shown in Fig. 2. While the result of the question ID 756 ranked the correct movie at 135 in the output ranking, that of the question ID 271 ranked the correct movie at $224,719$ (equivalent to 0.9692 of relative ranking). The amount of questions whose relative ranking is 0.2 or less is 52.0% of the total number of dev dataset.

[5] $[x]_+$ represents the function which returns 0 if $x < 0$ and returns x otherwise.

Table 3: Summary of the evaluated performance by using BERT

Type	Relative Rank	Question ID
Average	0.2543	–
Minimum	5.823×10^{-4}	756
First Quartile	4.717×10^{-2}	–
Median	0.1855	–
Third Quartile	0.3668	–
Maximum	0.9692	271

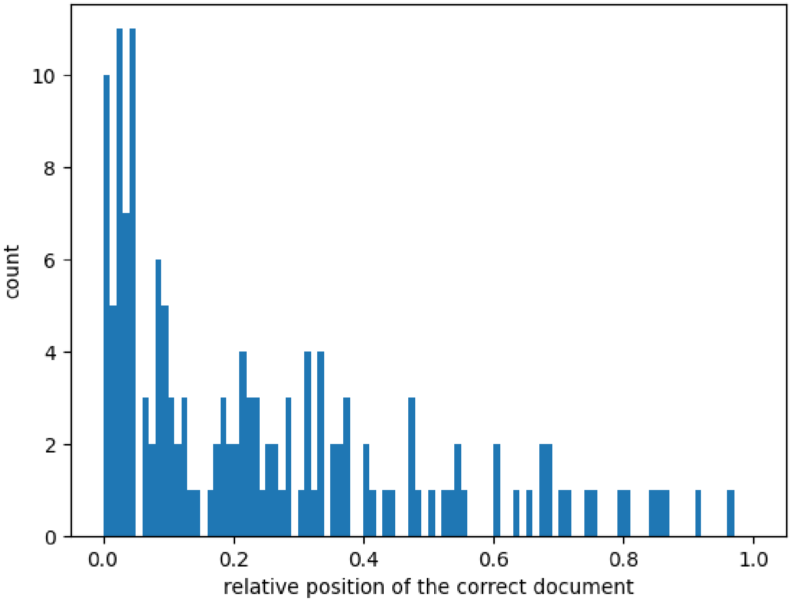

Fig. 2: Distribution of the relative rank of correct movies in outputs by using BERT.

In a preliminary experiment, we observed different tendencies in the results based on the TF–IDF and BERT models as shown in Fig. 3. Specifically, there were some queries for which the TF–IDF model outperformed BERT, and vice versa. Therefore, we expected the combination of the two models to perform well.

3.5 Hybrid Model

Next, we set up a hybrid model combining TF–IDF with BERT. In this experiment, we calculated the linear sum of the outputs of both TF–IDF and BERT to produce the final ranking. However, while the output of TF–IDF is the cosine similarity, that of BERT is the Euclidean distance, so we cannot combine them directly. Therefore, to combine these scores, we used Eq. (1) where s_t is the score

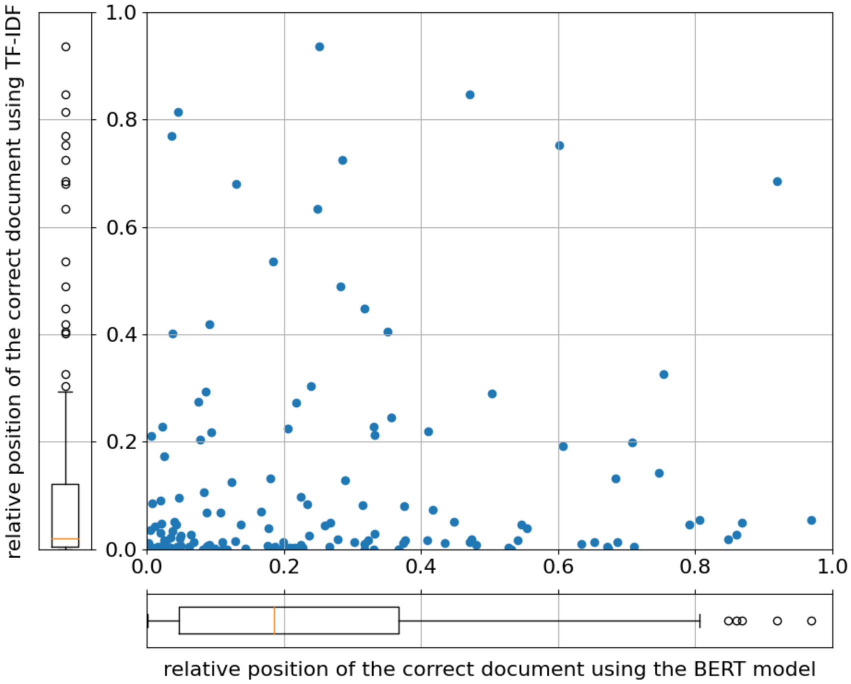

Fig. 3: Scatterplot to compare the system between TF–IDF and BERT. The vertical axis and the box plot shown on its outside represent the performance of BERT, and the horizontal axis and the box plot shown on its outside represent that of TF–IDF.

from the TF–IDF model, s_b is that from the BERT model, and both a and b are positive real numbers.

$$S = as_t + \frac{b}{s_b} \tag{1}$$

S represents the overall relevance of the movie to the query. The combined model generated the list of movies by sorting S in descending order.

The real numbers a and b that make the result of the dev dataset the best were determined by using a grid search. The result of the grid search is shown in Fig. 4 and it shows that the coefficient $a = 5.0 \times 10^{-3}$ and $b = 1.0 \times 10^{-4}$ is the best value as the relative rank is 0.1029. This indicates that the combined model is better than the two component models.

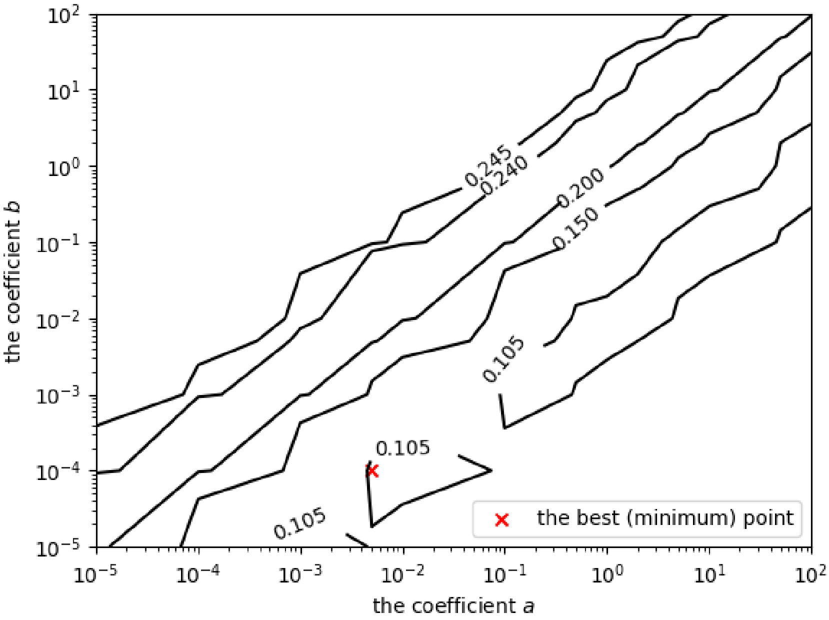

Fig. 4: Contour map that indicates the relation between the coefficients a, b and the relative rank of the result.

4 Discussion

We analyzed a few questions and their preprocessed Wikepedia texts for which our models performed particularly poorly. The worst result from the TF–IDF model is the question ID 861[6], and its overview is as follows.

Question text Details are provided so that we can tell what the characters go through and what happens in the end. In addition, the time of viewing the movie and the expected release year are provided. Hence the information seems sufficient to identify the movie sought.

Its preprocessed Wikipedia text The description of the plot is completely missing. It is only a short abstract that contains the producer/author credits and the release date.

As described above, there is not enough information to specify the correct movie, so in this example it seems impossible to identify using only the preprocessed Wikipedia text.

On the other hand, the worst result from the BERT model is the question ID 271[7] and its overview is as follows.

[6] https://irememberthismovie.com/husband-goes-insane-gets-eaten-by-a-worm-and-comes-back-as-a-cat/

[7] https://irememberthismovie.com/shopping-mall/

Question text Compared to the previous example, the question is vague, but describes the synopsis of the whole story. In addition, its description uses direct expressions about sexual scenes.

Its preprocessed Wikipedia text The information about movie producer and actors is well written in the summary section. Its synopsis is also well written in the "plot" section, but mainly uses indirect expressions about the sexual scenes.

As described above, it seems to have enough information because of the richness of the "plot" section but seems to use expressions quite different from the question. Furthermore, the temporal order of the question is also different from the preprocessed Wikipedia text. Therefore, it seems to be difficult to retrieve the correct movie.

According to the paper by Arguello et al.[2], 8.7% of the question contains incorrect information about the movies. To minimize the effect from such an incorrect information, splitting the question text into several parts may be a better way to retrieve the correct movie.

5 Conclusion

In this study, we focused on the situation that a user can only remember the synopsis of the movie, character traits, etc., but not the title of the movie, and constructed a movie retrieval system specialized for this situation. As a result, the average relative rank of the system using TF–IDF is 0.1131, and that of the system using BERT is 0.2543. Although BERT is inferior to TF–IDF in this result, as each system showed different tendencies in the result, we tried to combine these two systems to construct the hybrid model. We introduced two coefficients into the formula and looked for the value for the best result by using a grid search. As a result, the average relative rank of the hybrid model is 0.1029, which is better than the component models.

The hybrid model is superior to the single model, but the effect size is not as large. It seems that improving the size and quality of the train dataset or modifying the learning method leads to better results.

Even if a question text contains incorrect information, splitting the question text into a sentence to minimize its effect may be a better way to retrieve the correct movie. We leave this research direction for future work.

References

1. Arguello, J., Bhargav, S., Mitra, B., Diaz, F., Kanoulas, E.: TREC 2023 Tip-of-the-Tongue (ToT) Track, https://trec-tot.github.io//
2. Arguello, J., Ferguson, A., Fine, E., Mitra, B., Zamani, H., Diaz, F.: Tip of the Tongue Known-Item Retrieval: A Case Study in Movie Identification. In: Proceedings of the 2021 Conference on Human Information Interaction and Retrieval. pp. 5–14. CHIIR '21, Association for Computing Machinery, New York, NY, USA (Mar 2021). https://doi.org/10.1145/3406522.3446021, https://doi.org/10.1145/3406522.3446021

3. Brown, R., McNeill, D.: The "tip of the tongue" phenomenon. Journal of Verbal Learning and Verbal Behavior **5**(4), 325–337 (Aug 1966). https://doi.org/10.1016/S0022-5371(66)80040-3, https://www.sciencedirect.com/science/article/pii/S0022537166800403
4. Cheng, D., Gong, Y., Zhou, S., Wang, J., Zheng, N.: Person Re-identification by Multi-Channel Parts-Based CNN with Improved Triplet Loss Function. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1335–1344 (Jun 2016). https://doi.org/10.1109/CVPR.2016.149, https://ieeexplore.ieee.org/document/7780518
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. CoRR **abs/1810.04805** (2018), http://arxiv.org/abs/1810.04805
6. Elsweiler, D., Ruthven, I., Jones, C.: Towards memory supporting personal information management tools. Journal of the American Society for Information Science and Technology **58**(7), 924–946 (2007). https://doi.org/10.1002/asi.20570, https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.20570
7. Hoffer, E., Ailon, N.: Deep metric learning using Triplet network (Dec 2018). https://doi.org/10.48550/arXiv.1412.6622, http://arxiv.org/abs/1412.6622
8. Kaya, M., Bilge, H.Ş.: Deep Metric Learning: A Survey. Symmetry **11**(9), 1066 (Sep 2019). https://doi.org/10.3390/sym11091066, https://www.mdpi.com/2073-8994/11/9/1066
9. Schwartz, B.L., Metcalfe, J.: Tip-of-the-tongue (TOT) states: Retrieval, behavior, and experience. Memory & Cognition **39**(5), 737–749 (Jul 2011). https://doi.org/10.3758/s13421-010-0066-8, https://doi.org/10.3758/s13421-010-0066-8
10. Wang, J., song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., Wu, Y.: Learning Fine-grained Image Similarity with Deep Ranking (Apr 2014). https://doi.org/10.48550/arXiv.1404.4661, http://arxiv.org/abs/1404.4661