Movie Keyword Search Using Large-Scale Language Model With User-Generated Rankings and Reviews

Tensho Miyashita[1], Yoshiyuki Shoji[1,2][0000−0002−7405−9270], Sumio Fujita[3][0000−0002−1282−386X], and Martin J. Dürst[1][0000−0001−7568−0766]

[1] Aoyama Gakuin University,
Sagamihara, Kanagawa 252-5258, Japan
tensho@sw.it.aoyama.ac.jp, duerst@it.aoyama.ac.jp
[2] Shizuoka University,
Hamamatsu, Shizuoka 432–8011, Japan
shojiy@inf.shizuoka.ac.jp
[3] Yahoo Japan Corporation,
Chiyoda, Tokyo 102–8282, Japan
sufujita@yahoo-corp.jp

Abstract. The paper proposes a novel method for conducting keyword-based movie searches using user-generated rankings and reviews, by utilizing the BERT language model for task-specific fine-tuning. The model was trained on paired titles and reviews, enabling it to predict the likelihood of a movie appearing in a ranking that includes a particular keyword. An experiment using data from a reputable Japanese movie review site demonstrated that the method outperformed existing similarity-based approaches. However, some aspects, such as pooling methods, could be improved for accuracy.

Keywords: Movie Search · Online Review · BERT · Learning to Rank

1 Introduction

The shift towards on-demand and subscription-based movie services has changed how people select and watch movies. Traditional methods like going to a movie theater or video rental store limit the choices. Still, online platforms offer many options, making it challenging to select a movie based on specific preferences. Meanwhile, the changing attitudes towards movie-watching, such as viewing movies on computers and smartphones, often while multitasking, increased the demand for more efficient and personalized search technologies. These technologies should ideally consider not just metadata but also subjective elements like "beautiful scenery" or "flashy action," which are not typically included in conventional search algorithms.

We present a method that utilizes Bidirectional Encoder Representations from Transformers (BERT) to rank movies based on free queries, harnessing user-generated rankings and reviews. This approach is particularly tailored for user-generated ranking platforms where users can compile lists of up to ten of their favorite movies along with a descriptive title, as seen in features like the “Round-Up” function on Yahoo! Movies, Japan.

We hypothesize that these descriptive titles can function as search keywords and proceed to train a model that can correlate these titles with the movies listed, thereby enabling more nuanced, keyword-based movie searches. The method assesses the likelihood of a movie appearing in user-generated rankings that contain the specified keyword, offering a more personalized search experience.

Our algorithm first vectorizes ranking titles and all reviews using BERT, and then it trains the neural network model to calculate the relevance between the ranking title and reviews of a particular movie. Finally, the model calculates the probability that a given movie will appear in the rankings with a given keyword in the title.

We conducted a subject experiment to confirm the accuracy and effectiveness of this ranking. Participants were asked to evaluate how closely the movies in the rankings generated by the proposed method, and several comparison methods were related to the query.

2 Related Work

This research aimed to make movies searchable using word of mouth (eWOM). BERT and Learning to Rank were used as enabling technologies.

Our research used reviews posted on movie sites, a type of eWOM, to find movies that were close to what users were looking for. Several examples of this kind of item search focus on eWOM. Ramanand *et al.* [5] proposed a method for extracting “wishes” that indicated suggestions about products and services and purchase intentions from documents, such as reviews and buyer surveys. Similarly, this study used user review information to search for movies using free queries.

We used BERT to search movies by using their review. BERT is a Large Language Model (LLM) proposed by Devlin *et al.* [1] that enables context reading. There are examples of BERT applied to information retrieval. Yang *et al.* [8] proposed a method for adapting BERT to the ad hoc retrieval of documents. Yunqiu *et al.* [6] proposed a BERT model for legal case retrieval, BERT-PLI, that can retrieve from much longer queries than general queries. The task of retrieving Lithuanian text and audio documents using the query and search corpus provided by IARPA’s MATERIAL program revealed that this method enables more accurate retrieval than other methods. Since this research was conducted on movies, which are visual images, it is difficult to deal with the contents of movies in text; therefore, we used user-posted reviews. In addition, it is difficult to compare the similarity of review sentences and short queries because of the different natures of the sentences. Therefore, we used Learning to Rank to match queries and review sentences.

Our method is a type of Learning to Rank, an information retrieval technique that uses machine learning. Three main approaches to Learning to Rank [4] are pointwise, pairwise, and listwise methods. In this study, we used the pointwise method. As an example of retrieving documents belonging to a specific topic, Amir *et al.* [7] proposed a method that used BERT and Learning to Rank to retrieve evidence to support a claim. Yu *et al.* [9] proposed a method using Learning to Rank to find documents containing answers to a given question. Since the current research is concerned with retrieving an item (*i.e.,* a movie), we also discuss some examples of applying Learning to Rank to item retrieval. Shubhra *et al.* [2] proposed a method that applied Learning to Rank for product retrieval on an e-commerce site. Prior to this study, Kurihara *et al.* [3] proposed a method using Learning to Rank for movie retrieval based on review information. The current work solves the same task with a more modern method (*i.e.*, LLM).

3 Movie Keyword Search Using Review and LLM

This section introduces an algorithm to rank movies based on any keyword query. The procedure, illustrated in Figure 1, initiates by retrieving titles and movies from user-generated movie rankings.

3.1 Vectorizing Movies Using Reviews

Movie websites feature reviews by various users. This research posits that these reviews encapsulate the movie's attributes. Consequently, review sentences, as opposed to movie metadata or visuals, were used to vectorize a movie.

For vectorization, the text underwent preprocessing. Sentences were segmented using punctuation, and superfluous characters and symbols were discarded. The average pooling is used to create a fixed-length vector of 768 dimensions.

3.2 Formatting User-Generated Rankings as Training Data

To facilitate movie retrieval for specific queries, data linking queries to movies is essential. The trend of users sharing personalized movie rankings online is increasingly prevalent. Many movie review sites allow users to create and register lists of favorite movies (*e.g.*, IMDB's Watchlist, Yahoo! Movies' Round-Up). User-generated rankings often have titles like "The 10 best movies that make me cry". Our method focuses on the relationship between this ranking title and the movies that appeared in the ranking. As preprocessing, unnecessary words (*e.g.*, "my", "all-time", "movie") were removed from the ranking titles.

3.3 Learning the Relationship Between Movies and Words in the Ranking Title

Relevance between a query and a movie is computable by considering the ranking title as a keyword query. Here, a simple neural network was used to calculate relevance. When each movie and query has been represented as a vector of distributed representations, the proposed method trains a neural network as a binary classification task using these vectors, as shown in Figure1. The task is to combine vectors of movies and queries and then estimate whether the movie will likely appear in a ranking that includes the query in its title.

The input was a 1,536-dimensional vector: a combined pair of a 768-dimensional vector representing a movie and a 768-dimensional vector representing a ranking title. The output layer was binary since it performs a binary classification

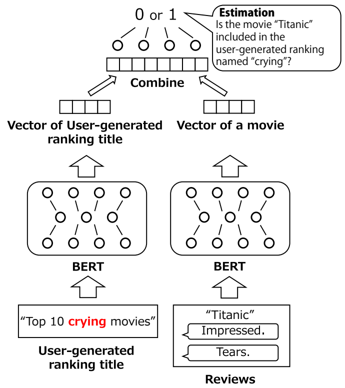

Fig. 1. Training that predicts whether a movie appears in user-generated rankings that include the query in the title.

Table 1. Queries used in the experiment and their features

Query	Features
Tearjerker Laughable Shocking	Emotion after watching the movie
Suitable for dating Suitable for children	Situation or scene when watching the movie
Suspense Animation	Category of the movie
Ghibli Surprise ending Takeshi Kitano	Content of the movie

of whether a movie appears in the ranking whose title contains a given query. Note that, the output value is 0 or 1 during training, but it takes the probability during the inference phase. We used this probability as the final ranking score.

This trained model can rank movies in response to any given query. The keyword was vectorized using BERT, combined with a movie vector, and fed into the trained model. The model predicts the probability of a movie appearing in user-generated rankings with that keyword. By applying this to all movies, movies can be ranked based on this probability.

4 Evaluation

A subject experiment was conducted to verify whether the movie ranking generated by the proposed method was consistent with the user's perception. We compared three variant methods and the proposed methods as follows:

- **Proposed Method** uses deep learning to estimate the relevance of the query vector to the vectors generated from the movie reviews.
- **Movie Similarity** compares the vectors generated from the movie reviews and the query vectors using cosine similarity. It uses the cosine similarity between vectors of the query and the movie as the ranking score.
- **Review Sentence Similarity** calculates cosine similarity between input queries and review sentences, using the most similar review for movies with multiple reviews. This comparative method tests the risk of compressing movie reviews with average pooling.
- **Metadata Only** uses only metadata and no reviews. It finds movies containing query keywords in their metadata, and it sorts them by cosine similarities between the query and description.

Table 2. Precision at k and nDCG for each method

	p@1	p@5	p@10	nDCG
Proposed Method	0.50	0.56	0.58	0.64
Metadata Only	0.50	0.46	0.40	0.60
Movie Similarity	0.20	0.24	0.27	0.47
ReviewSentenceSimilarity	**0.80**	**0.74**	**0.72**	**0.75**

Table 3. Example output of the proposed method worked well (query: "Tearjerker")

Rank	Movie title	Participant Rating
1	What Dreams May Come	3.0
2	Gray Sunset	3.5
3	Glory Daze	3.0
4	The Boy Who Could Fly	4.0
5	Crayon Shin-chan: The Adult Empire	4.5
6	Pay It Forward	3.5
7	It's a Wonderful Life	4.0
8	Jack	3.5
9	Life is Beautiful	4.0
10	The Notebook	4.0

4.1 Experiment

We prepared the training dataset consisting of 15,000 movies, and 10,000 user-generated rankings taken from Yahoo! Movies Japan. We prepared ten queries in advance (see Table 1).

A subject experiment was conducted online in which participants were asked to label the relevance between the query and the movie. Ten participants labeled the 5-point scale relevance between a query and 40 movies; the top ten movies of four methods. Participants can search for the movie if they do not know about it.

For the actual experiment, we implemented an actual movie search system. The BERT Japanese pre-trained model[4] was used. A Japanese morphological analyzer "MeCab[5]" and it's dictionary called "mecab-ipadic-neologd[6]" was used for a tokenizer.

4.2 Experimental Results

The precision for each method is shown in Table 2. In nDCG and p@k, the proposed method performed better than simple cosine similarity to metadata and review. The method that worked best was the method using the most similar sentences rather than the entire review. Examples of outputs when the proposed method worked well are shown in Table 3. Many "tearjerker" movies can be retrieved correctly.

5 Discussion

Overall, the proposed method was more accurate for retrieval than the cosine similarity or metadata methods. However, a simple similarity calculation be-

[4] Kurohashi Lab. Kyoto University: https://nlp.ist.i.kyoto-u.ac.jp/EN/
[5] MeCab: Yet Another Part-of-Speech and Morphological Analyzer: https://taku910.github.io/mecab/
[6] mecab-ipadic-NEologd: Neologism dictionary for MeCab https://github.com/neologd/mecab-ipadic-neologd

tween a single sentence in a movie review and a keyword query was even more accurate.

The method's limitation lies in representing a movie with a single vector, which might retain major trends like "sad" or "funny" but dilute minor impressions from individual scenes. Despite this, the proposed method outperformed superficial cosine similarity, likely because using a neural network better captures the nuanced differences between short keyword queries and longer, multi-person reviews than just cosine similarity does.

The Review Sentence Similarity method offers superior accuracy due to the detailed variance representation derived from review sentences. The highest nDCG was observed in Review Sentence Similarity, underscoring the value of using reviews. The results suggest that in our test search task, the presence of elements relevant to the query was more crucial than the general trend of the movie.

6 Conclusion

This paper presents a method to determine the relevance between keywords and movies using user-generated rankings and reviews. Experimental results showed that our method outperformed traditional cosine similarity, but in some tasks, using metadata was more accurate.

We are planning a more detailed evaluation and refinement of our methodology. We also intend to use this method for the fine-tuning of the model itself.

Acknowledgments

This work was supported by JSPS KAKENHI Grants Numbers 21H03775, 21H03774, and 22H03905.

References

1. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proc. of NAACL 2019. pp. 4171–4186 (2019)
2. Karmaker Santu, S.K., Sondhi, P., Zhai, C.: On application of learning to rank for e-commerce search. In: Proc. of SIGIR 2017. pp. 475–484 (2017)
3. Kurihara, K., Shoji, Y., Fujita, S., Dürst, M.J.: Learning to rank-based approach for movie search by keyword query and example query. In: Proc. of iiWAS 2021. pp. 137–145 (2021)
4. Liu, T.Y.: Learning to rank for information retrieval. Found. Trends Inf. Retr. pp. 225–331 (2009)
5. Ramanand, J., Bhavsar, K., Pedanekar, N.: Wishful thinking - finding suggestions and 'buy' wishes from product reviews. In: Proc. of NAACL 2010. pp. 54–61 (2010)
6. Shao, Y., Mao, J., Liu, Y., Ma, W., Satoh, K., Zhang, M., Ma, S.: Bert-pli: Modeling paragraph-level interactions for legal case retrieval. In: IJCAI. pp. 3501–3507 (2020)
7. Soleimani, A., Monz, C., Worring, M.: Bert for evidence retrieval and claim verification. In: Proc. of ECIR2020. pp. 359–366 (2020)
8. Yang, W., Zhang, H., Lin, J.: Simple applications of bert for ad hoc document retrieval. arXiv preprint arXiv:1903.10972 (2019)
9. Yu, L., Hermann, K., Blunsom, P., Pulman, S.: Deep learning for answer sentence selection. Proc. of NIPS 2014 (2014)