WUT21 at the NTCIR-16 Data Search 2 Task

Lin Li
Wuhan University of Technology
cathylilin@whut.edu.cn

Xinyu Chen
Wuhan University of Technology
268595@whut.edu.cn

Sijie Long
Wuhan University of Technology
sijie.long@whut.edu.cn

ABSTRACT

The WUT21 team participated in the IR subtask of the NTCIR-16 Data Search 2 Task. This paper reports our approach to solving the problem and discusses the official results.

Our approach aims to choose a simple base model, for the IR subtask, using a document-based storage method to facilitate retrieval of specified fields, thereby formulating a retrieval strategy. Elastic Search is a distributed full-text search and analysis engine based on Lucene, which has the advantages of high performance, high scalability, and real-time performance. Based on Elastic Search, the strategy uses embedded retrieval algorithms to retrieve topics and calculate text similarity, and select the optimal algorithm to match topic texts according to the final evaluation index NDCG@10. The final results show that the basic text similarity algorithm has a relatively high contribution performance for information retrieval tasks.

KEYWORDS

query expansion, text similarity, Elastic Search

TEAM NAME

WUT21

SUBTASKS

IR subtask (English)

1 INTRODUCTION

The WUT21 team participated in the IR subtask of the NTCIR-16 Data Search 2 [1]. This paper reports our approach to solving the problem and discusses the official results.The IR subtask is an information retrieval task based on the statistical data of the Bureau of Statistics. It matches documents according to topics, generates a ranking of documents, and finally returns the recommended sequence of documents and the corresponding scores.

For this IR subtask, you can see in the task overview that the task data is in JSON format, mainly text data. Therefore, the information retrieval task can be transformed into a sentence-level classification task. In this experiment, the sentence-pair classification method is used to calculate the similarity between the subject field and the file content, and return a descending order of similarity.

A common strategy for text-based information retrieval is to use a ranking function to rank all texts according to search terms and select the top n. And many text similarity algorithms are derived from the most basic word frequency similarity algorithm.

This report describes and discusses our results using different textual similarities for topics in the IR subtask to calculate how well topics match documents and returning a sorted list.

2 RELATED WORK

Traditional information retrieval tasks often use text classification methods to store data in the form of inverted indexes, and a common suggestion to users for coming up with good queries is to think of words that would likely appear in a relevant document, and to use Those words as the query. Finally return to the user a descending list of the query content, as many search engines do. The algorithm for calculating the score is usually a text similarity algorithm, such as the TF-IDF algorithm based on word frequency and inverse text frequency index[2].

For Elastic Search software, as a distributed, scalable, real-time search and data analysis engine, it is usually used to realize data search, analysis and exploration, or real-time statistics of structured data. In practice, Elastic Search is commonly used to complete the construction of a real-time search platform or the design of a retrieval system.

Due to the built-in controllable text similarity kernel in Elastic search, this experiment examines the possibility of its use in natural language processing tasks, and uses its retrieval module for text matching problems.

In this experiment, ES(Elastic Search) software and its embedded text similarity algorithms are used, such as: classic TF-IDF algorithm, BM25 algorithm, DFR similarity algorithm, DFI similarity algorithm, IB similarity algorithm, LM similarity algorithm.

3 METHODS

In the experiment, according to the results of NDCG@10 returned by the training set query, the final submitted version uses the text similarity return list calculated by the LM Jelinek Mercer Similarity algorithm. LM Jelinek Mercer Similarity is an algorithm embedded in ES (Elastic Search) software.

ES (Elastic Search) is based on the inverted index. By scanning each word in the article, an index is established for each word, indicating the number and position of the word in the article. When the user queries, the retrieval program searches according to the pre-established index, and feed back the search results to the user. Thus, a fast and efficient search function can be realized.

The retrieval function is of the form.

Under the query-likelihood approach, language models for IR try to estimate for each document the probability that the query Q was generated by the underlying language model, M_D. If it is assumed that terms occur independently, then the probability becomes the product of the individual query terms given the document model:

$$P(Q|M_D) = \prod_{t \in Q} P(t|M_D) \quad (1)$$

In information retrieval, it is common to use unigram models, where terms do not depend on their context. (While more sophisticated models could be expected to improve performance, work using higher order models has not been able to demonstrate consistent gains for IR, while such models are much more complex to estimate [4].) It therefore remains to estimate the probability of individual query terms. The document under consideration, D, is a sample from the language model, M_D. The maximum likelihood estimate of an individual query term is therefore given by:

$$\widehat{P}(t|M_D) = \frac{f_{d,t}}{|D|} \quad (2)$$

where $f_{d,t}$ is the within-document frequency of term t in document d, and $|D|$ is the total number of terms in the document.

The maximum likelihood language model M_C based on the term frequencies in the collection as a whole:

$$P(Q|M_C) = \prod_{t \in Q} P(t|M_C) \quad (3)$$

$$P(t|M_C) = \frac{l_t}{l_c} \quad (4)$$

l_t is the number of times the term shows up in the collection; l_c is the number of terms in the whole collection.

Jelinek-Mercer smoothing [Jelinek and Mercer, 1980] combines the relative frequency of a query term in the document D with the relative frequency of the term in the collection as a whole. The maximum likelihood estimate is moved uniformly toward the collection model probability $P(t|M_C)$:

$$P(Q|d,c) = \lambda \cdot P(Q|M_D) + (1-\lambda) \cdot P(Q|M_C) \quad (5)$$

The value of λ is query- and collection- dependent. A value of $\lambda \approx 0.1$ is suitable for short queries, and larger values (e.g. $\lambda = 0.7$) are more suitable for longer queries [6].

4 EXPERIMENTS

Based on the provided training set data, statistical analysis was performed on the training set data. The data format provided by the training set is Topic_id, Doc_id, and relevance label. The relevance labels are divided into L0, L1, and L2, which represent the relevance from low to high, respectively.

According to the standard of the NDCG evaluation index, the prediction of the L2 label occupies a large proportion in the score.

Table 1 and Table 2 summarize the basic situation of L2 labels in the training set.

Table 1 records the number of L2 labels in the training set and the size of the training set.

Table 1: The number of L2 labels in the training set.

	training set
L2 label	141
sum	10536

Table 2 records the number of topics with L2 labels and the total number of topics in the training set.

Table 2: The number of topics with L2 labels.

	Number of topics
L2 label	69
sum	192

Figure 1 counts the number of document entries in the training set corresponding to each topic.

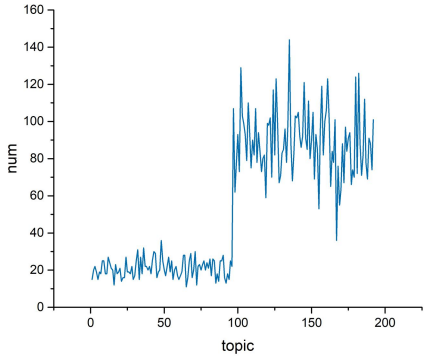

Fig.1. Training set distribution

Table 3 summarizes the label distribution and the label percentage of the training set.

Table 3: The label distribution of the training set.

	L2	L1	L0	sum
number	141	1434	8961	10536
percentage	1.34%	13.61%	85.05%	100%

Figure 2 counts the number distribution of L2 label in topics.

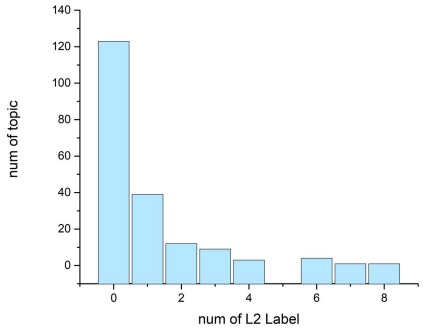

Fig.2. L2 label distribution

It can be seen from Table 3 that the proportion of L2 labels in the training set is very small, and a large amount of data is L0 labels. This means that when the similarity calculation method is selected according to the evaluation standard of NDCG@10 through the training set, the L2 label and the L1 label have a high weight. However, the results queried by ES(Elastic Search) do not contain labels, and only a query list with descending scores is returned.

When we use ES (Elastic Search) for retrieval, ES (Elastic Search) can customize the selection of specified fields for retrieval. As can be seen in the task overview [1], the fields contained in a document data, the fields related to the subject content are mainly the document subject and document content. Based on the training set, we use ES to compare three ways of retrieving only document subject or document content and retrieving document subject and document content simultaneously. According to the accuracy rate, the method of only retrieving document content is selected to be retrieved with ES (Elastic Search).

When calculating the training set NDCG@10, first obtain the returned list of each topic queried in the data set according to ES (Elastic Search), and then find the label in the training set through the ID of the document. If there is no query, write it as the L0 label.It is worth noting that in the data set, due to data redundancy, that is to say, documents with different IDs have the same content, so after the document ID is not queried, it is necessary to compare the content with the data in the training set. Finally complete the task of tagging.

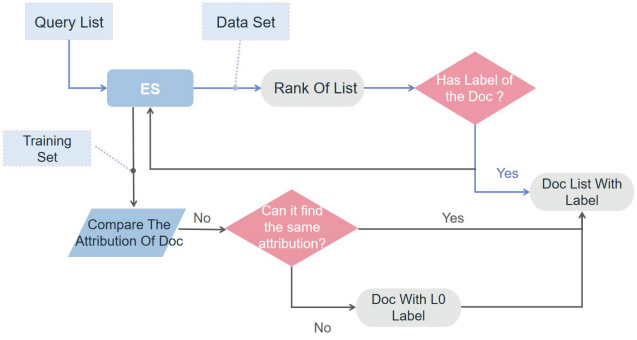

Fig.3. Search process

Finally, in the NDCG (main evaluation metric) of the test set, our performance is shown in Table 4.

Table 4: NDCG results.

	NDCG@3	NDCG@5	NDCG@10
score	0.1756	0.1661	0.1796

5 CONCLUSIONS

This report describes the results of the WUT21 team on the NTCIR-16 DataSearch-2 IR subtask. Judging from the results, this report still has a certain utility to query the subject words through the embedded LM similarity algorithm by using ES (Elastic Search) retrieval. This also shows that, in addition to using ES (Elastic Search) as a database as part of the system construction, ES (Elastic Search) can also be used as an information retrieval analysis tool by adjusting the parameters of the ES (Elastic Search) embedded similarity algorithm model.

In the final performance results, the effect presented by our team is moderate in the overall performance, so we also propose some possibilities for improvement based on the data set. For the data set, only two fields, document content and document subject, are used in this experiment, but looking at the original data set, we can see that there are some other fields in the data for text descriptions, such as data.data_organization and data_fields.tags Fields that describe the organizer of the text and the label of the text. In the follow-up work, you can try to use these fields to improve the retrieval effect. In addition, since the text similarity algorithm generally lacks the part that describes the related information of words, it is also possible to choose to add a neural network probabilistic language model as an improvement.

REFERENCES

[1] Makoto P. Kato, Hiroaki Ohshima, Ying-Hsang Liu, and Hsin-Liang Chen. 2022. Overview of the NTCIR-16 Data Search 2 Task. In *Proceedings of the NTCIR-16 Conference*.

[2] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008.ISBN:978-1-4503-0000-0/18/06 [J]

[3] Bennett G, Scholer F, Uitdenbogerd A. A comparative study of probabilistic and language models for information retrieval[C]//Database Technologies 2008: Proceedings of the Nineteenth Australasian Database Conference (ADC 2008). RMIT University, 2008: 65-74.

[4] P. Bailey, N. Craswell, and D. Hawking. Engineering a multipurpose test collection for Web retrieval experiments. Information Processing and Management, 39(6):853–871, 2003.

[5] W. Kraaij. Variations on Language Modeling for Information Retrieval. PhD thesis, University of Twente, June 2004.

[6] C. Zhai. Statistical language models for information retrieval, 2006.http://sifaka.cs.uiuc.edu/lmir/sigir06-tutorial-lmir.pdf(accessed 30 September 2006).

[7] Biadsy F, Nirschl M A, Ma M, et al. Approaches for neural-network language model adaptation[J]. 2017.