OUHCIR at the NTCIR-16 Data Search 2 Task

Satanu Ghosh
School of Library and Information Studies
The University of Oklahoma
U.S.A
satanu.ghosh-1@ou.edu

Jiqun Liu
School of Library and Information Studies
The University of Oklahoma
U.S.A
jiqunliu@ou.edu

ABSTRACT

In this paper, we report our work and discuss the results for NTCIR-16 DataSearch-2 IR subtask. NTCIR-16 Data Search-2 was organized to improve the present knowledge and promote the concepts of dataset search among IR researchers. In this particular subtask, we tried to perform ad-hoc retrieval for datasets based on given queries. While this task was available in English and Japanese, we decided to only compete for the English subtask. We sought to perform the ranking of datasets by using traditional BM25-based ranking functions and recent language models. During the evaluation experiments, we also explored the impact of metadata features on the performance of dataset ranking. Our best performing submission achieved a score of 0.153, 0.161 and 0.174 in nDCG@10, nERR@10, and Q-measure, respectively. In all the metrics, this run was ranked 13th among 25 submitted runs.

KEYWORDS

ad-hoc retrieval, dataset search, BM25, TF-IDF, doc2vec, SBERT

TEAM NAME

OUHCIR

SUBTASKS

IR Subtask (English)

1 INTRODUCTION

NTCIR-16 Data search 2 task [3] is an unique attempt to make progress in a new challenge in IR evaluation: ad-hoc retrieval of datasets. With the open data movement outreaching different parts of the world, several nations are encouraging citizen scientists by providing open access to datasets. Several web repositories can also be found that provide dataset access. While the influx of data can help scientists make data-dependent discoveries, it will also be a potential problem for search engines to rank datasets depending on user queries. This task aims to build upon the objectives of NTCIR-15 Data search task 1 [2], that were related to improving and overall understanding of data search following:

- Query understanding
- Data understanding
- Retrieval models

Data search can present several processing challenges, primarily because the task requires a pairwise comparison between the query and the document for (n X m) times if n is the number of queries and m is the number of datasets. Each dataset contains one or more documents. Therefore, if there are on an average p documents in each dataset then:

$$NumberOfComparisons = n(mXp)$$

Our approach was more exploratory because this was the first time we participated in this task. We wanted to start with simple, fast ranking models and then try more complex and time-consuming models. It is important to clarify that NTCIR-16 allowed participants to submit runs for both English and Japanese data search tasks, but we only participated in the English task. We did not participate in the Japanese track because our constraint of time and limited knowledge of the language and its properties. In Section 2 we will describe the resources that we used. Following this, we describe our Method (Section 3) for ranking the datasets. In section 4 we discuss our results and what they potentially mean. Lastly, in conclusion (Section 5), we will briefly summarize our work and discuss future direction.

2 RESOURCE DESCRIPTION

As we only participated in the English task, we will only provide details about the English Data Search-2 IR subtask resources.

There were 192 training queries and 58 test queries in the resource set. The pool of dataset contained 46,615 datasets with containing 92,930 documents of various types (.pdf, .xls, .json, .csv, .jpeg). Each dataset had a meta entry with different types of meta-information about the dataset; an example can be found in Figure 1. There were 10,536 annotated instances of query-document relevance labels provided for training purposes. The annotation was on gradient scale L0, L1, and L2, with L0 being irrelevant and L2 being highly relevant. More detailed description of the resources can be found in [3].

3 METHOD

Due to the length of the datasets and the potential problem of query-document asymmetry, we wanted to rank the datasets based on their meta information. We used the Title (title) and Description (desc) meta-information for each dataset for our entire experiment and did not analyze the documents listed inside them. We started our experiments with some fast and easy ranking functions like TF-IDF, BM25, and doc2vec and then tried to combine using fixed parameters to rank the datasets. Finally, we implemented a version of BERT called SBERT that can compute semantic similarity between two pieces of texts faster than traditional BERT. The details of each of the scoring functions, can be found in the following subsections:

3.1 TF-IDF

This is one of the simplest ways of measuring relevance between query and document by statistically identifying the terms that reflect more importance [9]. The mathematical equation can be found in 1

```
{
    "id": "0457e352-fb27-447e-9404-f82a16739ffb",
    "title": "Ruby Lake National Wildlife Refuge Narrative Report January - April 1961",
    "description": "This report for Ruby Lake National Wildlife Refuge outlines Refuge
    accomplishments from January through April of 1961. The report begins by summarizing the
    weather conditions, habitat conditions,  water conditions, and food and cover during this
    period. Wildlife- including migratory birds, upland game birds, big game animals,
    furbearers, predators, rodents, raptors, and fish- is also covered. The Refuge
    development and maintenance section discusses physical developments, plantings,
    collections and receipts, prescribed fires, and wildfires. Resource management is
    outlined; topics include grazing and haying. The public relations section of the report
    describes recreational uses, Refuge visitors,  Refuge participation, and hunting. Items
    of interest, N-R forms, and photographs are attached.",
    "data": [
    "data_fields": {
    "url": "https://catalog.data.gov/dataset/0457e352-fb27-447e-9404-f82a16739ffb",
    "attribution": "Ruby Lake National Wildlife Refuge Narrative Report January - April 1961 (
    https://catalog.data.gov/dataset/0457e352-fb27-447e-9404-f82a16739ffb) is licensed under
    U.S. Government Work (http://www.usa.gov/publicdomain/label/1.0/)"
}
```


Figure 1: Meta-information example

$$tf.IDF(t,d,D) = log(1+freq(t,d)).log(\frac{1}{count(d \in D, t \in D}) \quad (1)$$

where, t is the term, d is the document, and D is the set of documents. It is clear from the equation that the log normalization is performed on term frequency and inverse of document frequency, which means high term-frequency and low document-frequency will result in high relevance.

Using TF-IDF we computed a relevance score between the (query, title) and (query, desc) separately for each query in the test set.

3.2 BM-25

BM-25 [10] is a scoring function that can be used to compute a query-document relevance score based on the terms present in the query and the document. While similar to TF-IDF, this scoring function considers the length of the document and the saturation of term frequency. Many researchers still use this popular method as a baseline and first-level retrieval model for many ad-hoc retrieval tasks. The mathematical description of BM-25 can be found below:

$$score(q,d) = \Sigma_{i=1}^{n}.IDF(q_i)\frac{f(q_i,d).(k_1+1))}{f(q_i,d)+k_1.(1-b+b.\frac{|d|}{avdl})} \quad (2)$$

where q is query with n terms, d is the document, f(q $_i$, d) represents the number of terms from query present in the document, |d| is the length of document, avdl is the average length of document (in this case title and desc), IDF is the inverse document frequency of query term q_i and can be expressed by equation 3.

$$IDF(q_i) = log\frac{N-df_t+0.5}{df_t+0.5} \quad (3)$$

where df_t is the number of documents containing the term q_i and N is the total number of documents in the collection.

k_1 and b are free parameters there chosen for optimizing the scoring function. The k_1 and b were set to 1.5 and 0.75, respectively.

Like TF-IDF, we compute (query, title) and (query, desc) scores separately for each query in the test set.

3.3 doc2vec

doc2vec [4] is an extension of word2vec [5] that is often used for many NLP related tasks. The difference between word2vec and doc2vec is while word2vec is used to turn each word into a vector representation, doc2vec provides a cumulative embedding for a set of paragraphs or lines. We used doc2vec to convert each query, title, and description to vector representations using the Distributed Bag of Words configuration. It is important to note that document descriptions and titles had independent embedding space. We trained the doc2vec model for 100 epochs on the training qrels provided to us for title and desc separately. Then, we used these models to rank the documents of the 58 test queries based on cosine similarity between the (query, title) and (query, desc). doc2vec was used with the idea that it can better handle the word-sense disambiguation and effectively match queries with the contextual idea presented in the description and the title of the datasets.

3.4 Sentence BERT

In recent years, attention-based transformer architecture [12] have somewhat revolutionized machine learning. This type of architecture has been used to train many dense language models. As an example, BERT[1] has been used to achieve state-of-the-art performance for many NLP tasks. BERT has also been extended for many IR-related tasks [7, 11]. With the high number of parameters, it can understand and comprehend contextual references in a text. While BERT can be inefficient for computing pairwise semantic similarity, Sentence BERT (SBERT) [8] is a Siamese hierarchical network of two BERTs used to compute the semantic

similarity in a much faster way. Also, the availability of pretrained SBERT models for ad-hoc retrieval of MS-MARCO encouraged us to use them in a zero-shot configuration. The models that we used were: msmarco-distilbert-cos-v5 and msmarco-MiniLM-L12-cos-v5 [1]. We used some basic scoring functions like TF-IDF and BM25 in combination (OUHCIR-E-6) to screen the top 10000 datasets. After appending the Title and Description of all the datasets (10000), we used SBERT to generate their embeddings. The similarity between the query and dataset embedding was computed using a dot-product function. The re-ranking was done based on the assumed semantic similarity between the dataset (title+description) and the query of the 10000 datasets.

3.5 Ranking functions

We wanted to evaluate the influence of the two meta-entities title and description on ranking. Therefore, for TF-IDF, BM25, and doc2vec, we calculated relevance scores between the query and the datasets based on both titles and descriptions. Finally, we combined the scores by using different weights for the title and description scores. We did not combine doc2vec scores or SBERT scores because they are not term-based statistical measures. Before merging them (TF-IDF and BM25), we standardized each score (TFIDFtitle, TFIDFdescription, BM25title, BM25description) by subtracting the mean and dividing them by the standard deviation. Following this, we merged them based on some constant weights mentioned in equation 4.

$$score_w = \alpha(TFIDtitle + BM25title) + \beta(TFIDFdescription + BM25decription) \quad (4)$$

In the above equation, α and β are constants with values of 0.5 and 1.0 and alternated to emphasize title or description as a relevance measure without completely discounting the other. Another scoring function was used where both the constants were set to 1, which means equal importance for both. We also used a max-pooling approach by selecting the highest standardized score achieved by a document using TF-IDF and BM25 on dataset title and description and then ranking them based on this score. Essentially, we created a metric to prioritize the documents that have been perceived relevant by any of the scoring methods (BM25 and TF-IDF) based on either title or description. For example, there are two documents, D1 and D2. D1 has been scored as 0.8, 0.2, 0.3, and 0.4 by the four scoring functions, and D2 has been scored 0.4, 0.5, 0.5, and 0.6 by the four scoring functions. In this scenario, D1 will be prioritized over D2 because D1 has higher relevance with the query on one of the four metrics. Other submissions include title and description-based cosine similarity matching using doc2vec and two different pretrained SBERT models to perform semantic similarity-based ranking using dot product function.

A detailed description of our ranking functions used and the related submission number can be found in Table 1

4 RESULT & DISCUSSION

The evaluation was performed by the NTCIR-16 committee based on several traditional metrics of IR and can be found in Table 2. The primary evaluation metric is nDCG@10. On this metric, our best run (OUHCIR-E-5) received a score of 0.153. On other metrics, nERR@10 and Q-measure, this run reached the value of 0.161 and 0.174, respectively. On all the metrics our best run was ranked 13th among 25 submissions (taken from the result presented in [3] 3). As the annotated relevance measure was on a gradient scale, with L0 being irrelevant and L2 being highly relevant. Therefore, we can infer that a low nDCG score indicates that a large portion of top-ranked documents in the retrieved results was not relevant to the queries. Unlike nDCG, nERR provides a higher score if one of the top-ranked documents is highly relevant. Hence, low nERR means that no highly relevant documents were encountered in the top k-rank of the retrieved list. When we cross-verified our results with the distribution of annotation labels in the test-qrels, we found that the number of highly relevant (L2) datasets for the test queries was very low. In fact, most of the datasets were annotated irrelevant (L0). Hence, we can say that it is natural that the nDCG and nERR are low because IDCG and Cumulative Gain for many queries on annotation labels are 0.

While the results were not significant for an ad hoc retrieval task, we can draw crucial inferences from our experiments. It can be observed from the results that using the description of the datasets gives better results than the titles. Also, semantic ranking methods like SBERT and doc2vec are not useful for dataset ranking. Combining the calculated relevance by two scoring functions on every dataset's title and description, we received the best result. So, we can safely infer that neither title nor description of a dataset can be given higher importance while ranking.

The run receiving the best result in the IR subtask of Datasearch-2 in NTCIR-16 used a combination of BM25 and BERT. However, the question remains whether the time and computational resource spent to fine-tune a BERT model on the entire dataset (documents inside datasets) is worth the improved performance considering the second-best run used BM25 (without the documents inside datasets).

5 CONCLUSION

In this paper, we described and explained our approach for the English IR subtask of Datasearch-2 from NTCIR-16. The workshop is a special effort to understand and improve dataset retrieval for the IR community. We aimed to explore different dimensions of the data and the IR models for this task. We used four types of ranking functions, i.e., TF-IDF, BM25, doc2vec, and SBERT, and combined two (TF-IDF and BM25) using various methods. Our best-performing run was ranked 13th among 25 submissions. Our best performing run ranked the documents based on the highest score incorporating four different measures using two different scoring functions on two meta-information elements (title and description). From our experiments, we can conclude that: 1) that neither title nor description is more important than the other for measuring the relevance of a dataset, and 2) due to the unique nature of this task, semantic models and, in particular, pretrained SBERT on another popular ad-hoc retrieval data (MS-MARCO) [6] does not work for the dataset retrieval task. In the future, we would like to explore more detail about the queries used and how they could change the retrieval results.

[1] https://huggingface.co/sentence-transformers

Table 1: Descriptions of runs

Run	Description
OUHCIR-E-1	Ranking of datasets using OUHCIR-E-6+SBERT (msmarco-distilbert-cos-v5) dot product similarity between query and dataset (title + description)
OUHCIR-E-2	Ranking of datasets using OUHCIR-E-6+SBERT (msmarco-MiniLM-L12-cos-v5) dot product similarity between query and dataset (title + description)
OUHCIR-E-3	Ranking of datasets by combining TFIDF and BM25 standardized scores and setting α to 1 and β to 0.5 (more importance on title)
OUHCIR-E-4	Ranking of datasets by combining TFIDF and BM25 standardized scores and setting α to 0.5 and β to 1 (more importance on the description)
OUHICR-E-5	Ranking was done by selecting the maximum standardized score of a dataset (title and description) achieved on both TF-IDF and BM25 for each query
OUHCIR-E-6	Ranking of datasets by combining TFIDF and BM25 standardized scores and setting α and β to 1 (equal importance)
OUHCIR-E-7	Ranking of datasets using doc2vec model trained on query and dataset titles. The similarity was calculated using cosine function
OUHCIR-E-8	Ranking of datasets using doc2vec model trained on query and dataset description. The similarity was calculated using cosine function

Table 2: Results of OUHCIR

Run	*nDCG@3*	nDCG@5	*nDCG@10*	*nERR@3*	*nERR@5*	*nERR@10*	*Q-measure*
OUHCIR-E-5	0.12	0.138	0.153	0.11	0.142	0.161	0.174
OUHCIR-E-4	0.071	0.093	0.126	0.096	0.087	0.107	0.124
OUHCIR-E-6	0.071	0.093	0.126	0.096	0.087	0.107	0.124
OUHCIR-E-3	0.047	0.064	0.083	0.073	0.058	0.075	0.086
OUHCIR-E-1	0.021	0.022	0.025	0.016	0.03	0.032	0.035
OUHCIR-E-2	0.021	0.022	0.025	0.016	0.03	0.032	0.035
OUHCIR-E-8	0.012	0.013	0.019	0.011	0.013	0.017	0.022
OUHCIR-E-7	0.011	0.011	0.012	0.012	0.009	0.009	0.01

6 ACKNOWLEDGEMENT

This research was partially supported by the National Science Foundation (NSF) Award IIS-2106152. We extend our acknowledgement to a fellow member Mr. Ben Wang from the Human-Computer Interaction and Recommendation (HCIR) lab at the University of Oklahoma. Also, we want to thank the entire team of NTCIR-16 Datasearch task who responded to us whenever we had any questions regarding test collections.

REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. (2018).

[2] Makoto P Kato, Hiroaki Ohshima, Ying-Hsang Liu, and Hsin-Liang Chen. 2020. Overview of the NTCIR-15 data search task. In *Proceedings of the NTCIR-15 Conference*.

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

[4] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*. PMLR, 1188–1196.

[5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[6] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

[7] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375* (2019).

[8] 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 and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 3982–3992.

[9] Gerard Salton and Michael J McGill. 1986. Introduction to modern information retrieval. (1986).

[10] K Spark-Jones, S Walker, and SE Robertson. 2000. A Probabilistic Model of Information Retrieval: Development and Comparative Experiments–Part 1 and 2. *Information Processing and Management* 36, 6 (2000), 779–840.

[11] Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT Rediscovers the Classical NLP Pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 4593–4601.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

Table 3: Results for English IR Subtask as taken from [3]

Run	*nDCG@3*	nDCG@5	*nDCG@10*	*nERR@3*	*nERR@5*	*nERR@10*	*Q-measure*
NYUCIN-E-1	0.234	0.246	0.261	0.203	0.261	0.275	0.289
ORGE-E-2	0.191	0.188	0.211	0.199	0.222	0.233	0.248
ORGE-E-7	0.196	0.207	0.209	0.201	0.225	0.244	0.252
STIS-E-1	0.163	0.173	0.202	0.192	0.183	0.2	0.221
STIS-E-2	0.172	0.175	0.201	0.191	0.188	0.201	0.218
ORGE-E-4	0.152	0.163	0.191	0.186	0.177	0.193	0.21
ORGE-E-6	0.152	0.155	0.187	0.185	0.174	0.186	0.206
ORGE-E-3	0.147	0.159	0.187	0.173	0.171	0.192	0.212
ORGE-E-5	0.149	0.158	0.182	0.187	0.175	0.188	0.203
ORGE-E-8	0.144	0.155	0.181	0.175	0.166	0.18	0.194
wut21-E-1	0.176	0.166	0.18	0.101	0.207	0.211	0.226
ORGE-E-1	0.143	0.149	0.179	0.178	0.163	0.175	0.192
OUHCIR-E-5	0.12	0.138	0.153	0.11	0.142	0.161	0.174
OUHCIR-E-4	0.071	0.093	0.126	0.096	0.087	0.107	0.124
OUHCIR-E-6	0.071	0.093	0.126	0.096	0.087	0.107	0.124
OUHCIR-E-3	0.047	0.064	0.083	0.073	0.058	0.075	0.086
KSU-E-9	0.057	0.067	0.069	0.046	0.068	0.08	0.088
KSU-E-7	0.052	0.052	0.051	0.036	0.057	0.063	0.066
KSU-E-5	0.026	0.039	0.044	0.034	0.033	0.046	0.051
KSU-E-1	0.027	0.037	0.039	0.026	0.033	0.044	0.05
KSU-E-3	0.023	0.021	0.028	0.022	0.028	0.03	0.038
OUHCIR-E-1	0.021	0.022	0.025	0.016	0.03	0.032	0.035
OUHCIR-E-2	0.021	0.022	0.025	0.016	0.03	0.032	0.035
OUHCIR-E-8	0.012	0.013	0.019	0.011	0.013	0.017	0.022
OUHCIR-E-7	0.011	0.011	0.012	0.012	0.009	0.009	0.01