STIS at the NTCIR-16 Data Search 2 Task:
Ad-hoc Data Retrieval Ranking
with Pretrained Representative Words Prediction

Lya Hulliyyatus Suadaa
Department of Statistical Computing
Politeknik Statistika STIS
lya@stis.ac.id

Lutfi Rahmatuti Maghfiroh
Department of Statistical Computing
Politeknik Statistika STIS
lutfirm@stis.ac.id

Muhammad Luqman
Department of Statistical Computing
Politeknik Statistika STIS
luqman@stis.ac.id

Isfan Nur Fauzi
Department of Statistical Computing
Politeknik Statistika STIS
isfan@stis.ac.id

ABSTRACT
In this paper, we present the system and results of The STIS team for the Information Retrieval (English) subtasks of the NTCIR-16 Data Search 2 Task. The data collections in this task consist of a pair of metadata and a set of data files. We only used title, description, and tags of metadata as input documents of our proposed approach to retrieve a rank of query-related data files. We proposed using a pre-trained model to capture representative words prediction for each document then calculate the similarity between the query and the representative words as a rank score.

TEAM NAME
STIS

SUBTASKS
IR Subtask (English)

KEYWORDS
data retrieval, pre-trained model, representative word prediction

1 INTRODUCTION
The Open Government Data (OGD) policies encourage the distribution of government datasets to be explored by the citizen to solve any tasks. More than half of datasets in Google Dataset Search, one of the largest dataset search engines on the Web, arrive from the US Government Open Data portal (data.gov) [1]. In dataset search, the users submit their information needs using a query then the results are retrieved based on the similarity of the query to the metadata published about the datasets [2].

The STIS team participated in the Information Retrieval (English) subtasks of the NTCIR-16 Data Search 2 Task [4]. In this subtask, we have to generate a ranked list of statistical datasets for each query from data collections of the US government (data.gov). We explored metadata of the datasets consisting of title, description, and tags as input documents of our retrieval approach. Following Ma et al. [5], we proposed using a pre-trained Transformer model to capture representative words prediction for each document. The similarity between the query and the representative words was calculated as a rank score of the candidate retrieved dataset.

2 DATASETS
The datasets used in this paper consist of:

(1) Dataset collections
The collections were published by the US government1 containing 46,615 datasets with 92,930 data files in various formats such as Excel (i.e., xls and xlsx), CSV, JSON, XML, RDF, PDF, and text files. Each dataset was supplemented by metadata that describe data information, including title, description, data format, data url, created date, and tags.

(2) Queries
Information needs were extracted from questions posted in community question answering such as “What are the largest causes of death in the United States in 1999-2016?”. Queries for answering information needs were collected by asking ten crowdsourcing workers. The query example of previous information needs is "causes of death us 1999-2016".

3 PRETRAINED REPRESENTATIVE WORDS PREDICTION
The representative words idea is inspired by a query likelihood model that ranks the documents based on the relationship between a query and the document contents. A query consists of terms that are likely to appear in documents representing the representative words that discriminate the ideal documents from others.

Pre-trained models have been successfully applied in many downstream natural language processing tasks, including information retrieval. We implemented Pre-training with Representative wOrds Prediction (PROP) for ad-hoc retrieval [5] in constructing the representative words prediction for each document. The architecture of the proposed ranking approach with pretrained representative words prediction can be seen in Figure 1.

3.1 Representative Word Sets Sampling
We sample a pair of representative word sets from vocabulary $V = \{w_i\}_{i=1}^n$ based on document language model following dirichlet distribution with probability $P(w_i|D)$ for word $w_i$ and document $D$. Query likelihood score function $QL(w_i, D)$ were calculated to each

1https://data.gov
We used hinge loss function for a pairwise loss of positive and negative words, which takes advantage of the pre-trained contextual embeddings.

\[ \text{loss} = \max(0, 1 - P(w_{\text{pos}}|D) + P(w_{\text{neg}}|D)) \]

We used hinge loss function for a pairwise loss of positive and negative words, as follows:

\[ \text{loss} = \max(0, 1 - P(w_{\text{pos}}|D) + P(w_{\text{neg}}|D)) \]

**3.2 Representative Words Prediction**

Using pairs of positive and negative words from the previous step, we finetune pre-trained Transformer BERT for representative words prediction task. Adopting the finetuned BERT approach in question answering task [3], we preprocess positive and negative word tokens and document tokens as input by inserting two special tokens, [CLS] and [SEP]. The [CLS] token is added to the beginning of input, and the [SEP] token is inserted after the query token to separate the representative words and document segments. The hidden state was obtained as follow,

\[ h_{\text{CLS}} = \text{Transformer}([\text{CLS}] + w_{\text{rep}} + [\text{SEP}] + D + [\text{SEP}]) \]

where \( w_{\text{rep}} = \{w_{\text{pos}}, w_{\text{neg}}\} \). Then, we compute the probability of word represent to the the document, as follows:

\[ P(w_{\text{rep}}|D) = \text{MLP}(h_{\text{CLS}}) \]

We used hinge loss function for a pairwise loss of positive and negative word representation, as follows:

\[ \mathcal{L} = \max(0, 1 - P(w_{\text{pos}}|D) + P(w_{\text{neg}}|D)) \]

**3.3 Rank Score Calculation**

The k list of representative word prediction \( \{w_{\text{prop}}\}^k \) from the previous finetuned model were used as representative words of documents. We obtain the rank score by calculating the context similarity between the query and \( \{w_{\text{prop}}\}^k \). We used BERTScore [6] which takes advantage of the pre-trained contextual embeddings from BERT and computes the similarity of words in query and documents by cosine similarity.

\[ S_{\text{prop}} = \text{avg}(\text{BERTScore}(\text{query}, w_{\text{prop}})), i = 1...k \]

As the second model, we also try to combine the base score of traditional information retrieval model (e.g. BM25) and \( S_{\text{prop}} \) as follows:

\[ S = (1 - \alpha)S_{\text{base}} + \alpha S_{\text{prop}}. \]

where \( \alpha \) is the weight of the relevance score using representative words.

**4 RESULTS**

The overall performances of our proposed ranking approach using pre-trained representative words prediction (PROP) and BERTScore are shown in Table 1. We can see that our ranking mechanism using the Finetuned BERT of representative words predictor (PROP) were not outperformed the traditional information retrieval of BM25. However, a slightly better nDCG@10 score of the re-ranking mechanism could be a good sign of the representative words prediction effect.

The samples of representative words prediction using the fine-tuned model in NTCIR-16 dataset were shown in Table 2.

**5 CONCLUSIONS**

In this paper, we proposed an ad-hoc retrieval approach for governmental statistical data. We used metadata of data files as document features consisting of title, description and tags. We proposed using a pre-trained model to capture representative words prediction for each document then calculate the similarity between the query and the representative words as a rank score. We also combined the representative similarity score to re-rank candidate documents of BM25 model for each query.

**REFERENCES**

<table>
<thead>
<tr>
<th>Model</th>
<th>nDCG@3</th>
<th>nDCG@5</th>
<th>nDCG@10</th>
<th>nERR@3</th>
<th>nERR@5</th>
<th>nERR@10</th>
<th>Q-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM25</td>
<td>0.191</td>
<td>0.188</td>
<td>0.211</td>
<td>0.199</td>
<td>0.222</td>
<td>0.233</td>
<td>0.248</td>
</tr>
<tr>
<td>Ranking by PROP and BertScore</td>
<td>0.172</td>
<td>0.175</td>
<td>0.201</td>
<td>0.191</td>
<td>0.188</td>
<td>0.201</td>
<td>0.218</td>
</tr>
<tr>
<td>BM25 + Reranking by PROP and BertScore</td>
<td>0.163</td>
<td>0.173</td>
<td>0.202</td>
<td>0.192</td>
<td>0.183</td>
<td>0.200</td>
<td>0.221</td>
</tr>
</tbody>
</table>

Table 1: Results from NTCIR-16 Data Search 2. The best score is in bold.

<table>
<thead>
<tr>
<th>Query</th>
<th>Representative Words</th>
<th>Rank_{BM25}</th>
<th>Rank_{PROP}</th>
</tr>
</thead>
<tbody>
<tr>
<td>causes of death us 1999-2016</td>
<td>x, county, cause, numerous, rate, new, census, update, scientifically, deaths, death, estimating, internationally, direction, mortality, death, poisoning, low, ages, rates, base, longer, meets, deaths, drug, death, computer, selected, estimated, coded, affect, states, death, poisoning, pending, ratings, x, poisoned, deaths, finalization, deaths, provisional, death, provisional, comparisons, classifications, updated, deaths, reported, death, deaths, causative, categories, specifically, cause, drug, counting, drugs, provisional, drug, delay, provisional, drug, x, updates, vital, drug, numbered, pending, drug, rated, rate, x, nchs, differing, rated, rating, census, causes, estimate, number, baseline, poisoned, base, acquisition, references, death, poisoned, drug, includes, update, cdc, adjusted, www, published, demographic, x, rating, ageing, death, defined, wonder</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>annual turnover care workers</td>
<td>disabled, incoming, benefits, statistics, insurance, social, socially, calculations, workers, securing, three, three, receiving, edition, calculate, disability, series, people, disability, tables, workers, peoples, social, statistical, low, researchers, teams, v, u, shorebird, shores, forage, forest, v, tennessee, foraging, group, radio, forest, primary, radio, migrate, ponds, estimating, migratory, shorebird, opening, vulnerable, estimated, network, rates, calidris, western, flight, shorebird, reservation, created, migration, establishing, rate, build, regional, valleys</td>
<td>56</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2: Samples of representative words prediction for each query and their ranks.


