HW5

Title: Learning Elasticsearch

Author: Tongkai Zhang

Date: Apr 21, 2021

Description: Implemented an IR system with Flask and Elastcsearch, providing a command line tool to process TREC queries over four type of search heuristics. Evaluated the result with NDCG@20 metric and also implemented a UI for interactive query processing.

System Overview

Analyzer

Customized: snowball stemmer, asciifolding token filter, lowercase filter, standard tokenizer

Matching and Ranking

This system first matches the query with related docs using BM25 algorithm. Based on the user's choice, the first 20 results can be reranked by the embedding algorithms.

Understandings of Embedding

main idea: turn the term to a dense real value vector, which can be learned from a given corpus. The vector contains the context of the word, so similar word would have higher score in cosine similarity.

IR: mean vector is calculated both for query and document. Matching score can be calculated from these two mean vectors

NDCG Score and Result

First Five Queries in TREC: 321 336 341 347 350

Bash File `evaluation.sh`

Query Example

```
python evaluate.py --index_name wapo_docs_50k --topic_id 321 --query_type title --top_k 20
python evaluate.py --index_name wapo_docs_50k --topic_id 321 --query_type title -u --top_k 20
python evaluate.py --index_name wapo_docs_50k --topic_id 321 --query_type narration --vector_name sbert_vector --top_k 20
python evaluate.py --index_name wapo_docs_50k --topic_id 321 --query_type title --vector_name sbert_vector --top_k 20
```

Topic 321

Query Type	title	description	narration
BM25 + default	0.813	0.684	0.626
BM25 + custom	0.631	0.780	0.584
fasttext + default	0.687	0.635	0.530
sbert + default	0.713	0.651	0.690

Analysis: For this topic, the best performance is reached by BM25 + default analyzer on the query type title, and embedding ranking only makes little improvement for performance score on narration. We may conclude that the title is informative enough for the system to match the expected result.

Topic 336

Query Type	title	description	narration
BM25 + default	0.823	0.358	0.430
BM25 + custom	0.845	0.431	0.381
fasttext + default	0.471	0.410	0.370
sbert + default	0.680	0.416	0.394

Analysis: The best performance is reached by BM25 + default analyzer on the query type title, and embedding ranking only makes little improvement for performance score on description. We may also conclude that the title is informative enough for the system to match the expected result.

Topic 341

Query Type	title	description	narration
BM25 + default	0.762	0.593	0.900
BM25 + custom	0.806	0.816	0.947
fasttext + default	0.805	0.619	0.752
sbert + default	0.780	0.688	0.763

Analysis: Best performance is reached by narration based search by BM25 + customized analyzer. Thus, this topic needs more context to be translated correctly. Customized analyzer with more normalization techniques can improve the performance for this topic

Topic 347

Query Type	title	description	narration
BM25 + default	0.350	0.468	0.323
BM25 + custom	0.235	0.432	0.503
fasttext + default	0.397	0.265	0.282
sbert + default	0.339	0.296	0.364

Analysis: Best performance is reached by narration based search by BM25 + customized analyzer. Similar conclusion to the previous topic can be drawn.

Topic 350

Query Type	title	description	narration
BM25 + default	0	0	0.356
BM25 + custom	0	0	1.0
fasttext + default	0	0	0.289
sbert + default	0	0	0.631

Analysis: This topic has zero title and description NDCG score, but quite high score got by narration with BM25 + customized analyzer. So the title and description is not informative enough for the topic, whereas the narration contains most important information for this topic.

Note

cosine similarity correction directory

```
/Applications/anaconda3/envs/cosi132a/lib/python3.7/site-packages/elasticsearch_dsl
```

Time Consumed: 16hr