A Report on

Vector Space Based Ranked Retrieval System

Submitted in fulfilment of the course
CS F469 - Information Retrieval

By
Vishal Mittal- 2017A7PS0080P
Yash Vijay- 2017A7PS0072P
Laksh Singla- 2017A7PS0082P

To
Dr. Abhishek,
Assistant Professor,
Computer Science and Information Systems Department
BITS-Pilani, Pilani Campus

11th March 2020

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

Introduction

We built a vector-spaced based information retrieval system, that takes in free text English language queries and produces a list of 10 (or K) documents that it finds most relevant. The *lnc.ltc* scoring scheme (based on SMART notation) was used to calculate scores for each document with the query. The IR system was tested on 10 different multi-term queries, and the top 10 documents were evaluated manually to find out the performance of the IR system.

Evaluation of the IR system

Query 1	Top 10 documents	Score	Relevant? Yes/No
Number theory	teiji takagi	0.1452	Yes
	faltings's theorem	0.1227	Yes
	victor vroom	0.1215	No
	goro shimura	0.1153	Yes
	abelian extension	0.1102	Yes
	abc conjecture	0.1084	Yes
	surautomatism	0.1081	No
	canterbury college	0.1069	Yes
	human development theory	0.1067	No
	planck temperature	0.1058	Yes

Query 2	Top 10 documents	Score	Relevant? Yes/No
Nazi germany	nuremberg rally	0.1088	Yes
	heroldo de esperanto	0.1084	Yes

	great patriotic war (term)	0.1062	Yes
	orwo	0.1021	Yes
	lower franconia	0.0938	Yes
	walther funk	0.0907	Yes
	inside the third reich	0.0868	Yes
	nord-fron	0.0840	Yes
	rhin	0.0813	Yes
	elisabeth bergner	0.0797	Yes

Query 3	Top 10 documents	Score	Relevant? Yes/No
Information retrieval	traceability	0.0514	Yes
	datenschlag	0.0404	Yes
	list of psychological research methods	0.0400	Yes
	mimer sql	0.0394	Yes
	list of boogie woogie musicians	0.0379	No
	personal information manager	0.0371	Yes
	rosetta (spacecraft)	0.0346	No
	romanization	0.0327	Yes
	esperanto association of britain	0.0281	No
	criminal justice: a brief introduction	0.0272	No

Query 4	Top 10 documents	Score	Relevant? Yes/No
Roman empire	579 bc	0.2319	Yes
	reign	0.1539	Yes
	hoeselt	0.1529	Yes
	list of imperial diet participants (1792)	0.1484	Yes
	1930 british empire games	0.1210	No
	limbourg	0.1198	No
	joannes	0.1146	Yes
	treaty of ryswick	0.1066	Yes
	1950 british empire games	0.1055	No
	maasmechelen	0.1040	No

Query 5	Top 10 documents	Score	Relevant? Yes/No
artificial intelligence and machine learning	probert encyclopaedia	0.1048	Yes
	inductive bias	0.0990	Yes
	max tegmark	0.0849	Yes
	perceptron	0.0842	Yes
	list of tunnels in new zealand	0.0728	No
	bütgenbach	0.0721	No
	extropianism	0.0671	No
	simearth	0.0615	Yes
	overfitting	0.0615	Yes

	oramics	0.0614	No

Query 6	Top 10 documents	Score	Relevant? Yes/No
Film industry	19th century in film	0.3292	Yes
	1912 in film	0.2574	Yes
	1911 in film	0.2574	Yes
	1910 in film	0.2574	Yes
	1909 in film	0.2574	Yes
	1908 in film	0.2574	Yes
	1907 in film	0.2574	Yes
	1906 in film	0.2574	Yes
	1905 in film	0.2574	Yes
	1904 in film	0.2574	Yes

Query 7	Top 10 documents	Score	Relevant? Yes/No
Fast food	junk food news	0.1226	Yes
	teriyaki	0.0846	Yes
	acer saccharinum	0.0794	Yes
	dilberito	0.0761	Yes
	pan (newsreader)	0.0729	No
	yorick (programming language)	0.0727	No
	simmering	0.0725	Yes

	populus sect. populus	0.0723	No
	cracker (food)	0.0686	Yes
	cooking show	0.0670	Yes

Query 8	Top 10 documents	Score	Relevant? Yes/No
Computer networks	turing (disambiguation)	0.1451	Yes
	smurf attack	0.1363	Yes
	warhol worm	0.1085	Yes
	keygen	0.1047	Yes
	personal information manager	0.0986	Yes
	network news transfer protocol	0.0975	Yes
	tcpdump	0.0973	Yes
	timeline of computer viruses and worms	0.0973	Yes
	franklin c. crow	0.0896	Yes
	eqp	0.0882	Yes

Query 9	Top 10 documents	Score	Relevant? Yes/No
Computer network	turing (disambiguation)	0.1740	Yes
	smurf attack	0.1642	Yes
	list of programs broadcast by upn	0.1412	Yes
	application layer	0.1303	Yes

	bitnet relay	0.1283	Yes
	tcpdump	0.1248	Yes
	1968 in television	0.1232	No
	1965 in television	0.1232	No
	1961 in television	0.1222	No
	1960 in television	0.1222	No

Query 10	Top 10 documents	Score	Relevant? Yes/No
Mahatma Gandhi	robert hart	0.0761	Yes
	meenachil	0.0638	No
	british museum reading room	0.0534	Yes
	nonviolence	0.0488	Yes
	moral example	0.0450	Yes
	missionary generation	0.0417	Yes
	palai	0.0410	No
	thrissur	0.0368	No
	bayard rustin	0.0310	Yes
	anarcho-pacifism	0.0299	Yes

From the 100 evaluated instances, 77 were found to be correct. Thus, the accuracy is 77%.

Implementation Details

For lemmatization, `WordNetLemmatizer` from nltk is used. Weighting scheme for ranked retrieval is lnc.ltc:

1. Before asking any queries the system pre-calculates term-document weights, using the formula 1 + log10(term_frequency) and normalizes it by document vector's length (for cosine similarity). Results are stored in a Dictionary for fast future accesses. Also, inverse document frequency (idf) is computed for all terms.
2. When free-text query is typed, the system computes term-query weights using formula (1 + log10(term_frequency_in_query)) * idf(term) and normalizes them. It requires linear time depending on the query length.
3. To efficiently calculate document scores, term-at-a-time approach (bag of words) is used for query terms:

   ```
   for term, query_weight in term_query_weights.items():
       for doc_id, doc_weight in term_doc_weights[term].items():
           doc_id_score[doc_id] += query_weight * doc_weight
   ```

   Time complexity will linearly depend on the number of term-document pairs for query terms.

4. Documents are sorted by their scores in (*O(N log N)*, where *N* is the number of documents, containing query terms) to show top relevant.

Improvements

We suggest two improvements for the given system, one to improve the accuracy, and the other one to improve the speed of execution.

- Lemmatization + Spelling Check : We used lemmatization and spell checker on our document and queries to ensure relevant documents are fetched independently of the grammatical form of words used in the queries and documents. Through this, we aimed to increase the relevance of the documents fetched.
- Champion List: We tried to increase the speed of execution by precomputing a list of n = 100 documents which have the highest weight for a particular term per their term frequency. Through this, we aimed to increase the speed of execution by avoiding the computation of all document rankings at query time.

Results of improvements:

- Before implementing spell correction, if we searched for a term with a spelling mistake, that term was simply ignored, for example: indusry. No results were produced. If we searched for film indusry, the results were:

```
score = 0.3292, document_id = 172582, title = 19th century in film
score = 0.2574, document_id = 172507, title = 1912 in film
score = 0.2574, document_id = 172509, title = 1911 in film
score = 0.2574, document_id = 172510, title = 1910 in film
score = 0.2574, document_id = 172511, title = 1909 in film
score = 0.2574, document_id = 172513, title = 1908 in film
score = 0.2574, document_id = 172514, title = 1907 in film
score = 0.2574, document_id = 172515, title = 1906 in film
score = 0.2574, document_id = 172516, title = 1905 in film
score = 0.2574, document_id = 172518, title = 1904 in film
```

The system searched only for film. After the spell check, the results are much better:

```
score = 0.1014, document_id = 172928, title = bopet
score = 0.1005, document_id = 173658, title = 110 film
score = 0.0974, document_id = 173656, title = 126 film
score = 0.0866, document_id = 178751, title = toronto international film
festival
score = 0.0803, document_id = 174051, title = edwin catmull
score = 0.0770, document_id = 171058, title = chaplin (film)
score = 0.0769, document_id = 175223, title = dx encoding
score = 0.0768, document_id = 173720, title = disc film
score = 0.0739, document_id = 175627, title = single-8
score = 0.0730, document_id = 170194, title = the dam busters (film)
```

- Consider queries 8 and 9 given above. A small change in the word network and networks resulted in very different results. After lemmatization, the results are as follows:

```
score = 0.1604, document_id = 176472, title = smurf attack
score = 0.1248, document_id = 176292, title = personal information
manager
score = 0.1239, document_id = 171062, title = bitnet relay
score = 0.1213, document_id = 170533, title = tcpdump
score = 0.1201, document_id = 173452, title = egon zakrajšek
score = 0.1200, document_id = 173004, title = network news transfer
protocol
score = 0.1085, document_id = 173921, title = gift
score = 0.1072, document_id = 174761, title = timeline of computer
viruses and worms
score = 0.1051, document_id = 174058, title = franklin c. crow
score = 0.1046, document_id = 171484, title = chinese social relations
```

- Lemmatization slows down the query search by a large factor since all the words (over 1 million) in the entire corpus are lemmatized one-by-one but the use of champion list over that improves the overall efficiency as now the number of documents searched in the worst case are CHAMPION_LIST_COUNT * (Number of query terms) which gives very fast results as expected and seen.