Saarland University

Statistical Natural Language Processing

Final Project: Information Retrieval

Authors:
Hassan Soliman 2576774

Lecturer: Univ.-Prof.
Dr.-Ing. Dietrich Klakow

Date of submission
Oct 4th, 2020

Contents

1 Introduction

Information retrieval is finding material - usually documents - of an unstructured nature - usually text - that satisfies an information needed from within large collections[1]. It tries to match these documents with the provided information in the query using different metrics, models and concepts. These methods vary and are evaluated according to the degree of relevancy of resulted documents to the query.

2 Problem Description

2.1 Problem Definition

Problem is defined as the task to implement a system that takes an input (query) and returns certain number of the most relevant documents with their rank. Afterwards the most relevant sentences from these documents are ranked and retrieved. The goal is to get the answer to the query from one of the top ranked documents and sentences.

2.2 Challenges

Generally, in the field of information retrieval, it's hard to get high scores, but in this project, the low scores can be attributed to the following points which represent challenges for the project:-

- Low Number of Documents:- The documents database is trimmed and only contains 8749 documents which is very low number to search from.

- Complexity of Queries:- Some queries are tricky and it can easily trick TF-IDF model as they contain the same terms as the ones exist in some documents, but without the relevant answer.

- Simplicity of Models:- Models used like TF-IDF model and ranking methods based on it e.g. BM25 are considered simple for the given queries and low number of documents.

- Missing Relevant Documents:- A few queries don't have relevant documents when executing matching of their answer patterns with the documents database.

3 Data Analysis & Preprocessing

3.1 Exploratory Data Analysis

The data files presented and explored are as follows:-

- Documents Data:- Documents are given in marked up language with tags. Two tags are important and extracted which represent the (Document No.) and (Document Text) respectively. Documents Numbers can represent IDs for the documents, while document text is a plain string which has the relevant content of the document.

- Query Data:- Queries are the same as documents in the since of being represented in marked up language with tags. Two tags are important and extracted which represent the (Query No.) and (Query Description) respectively. Query descriptions represent plain string which has the relevant question of the query.

- Patterns Data:- Patterns for the answers to the queries are given in a plain text file. Each query number has the corresponding answer patterns in this file. It represents as the ground truth for the queries. For each query, the corresponding patterns are matched with the documents to get the relevant documents for that query.

3.2 Data Preprocessing

Data is prepared and preprocessed as follows:-

- Documents Data & Query Data:- Documents and Queries are preprocessed the same. First punctuation marks are removed, then texts are lower-cased. Finally tokenisation is applied.

- Patterns Data:- Patterns are extracted for each query. Without preprocessing, it's matched with the documents texts without preprocessing as well.

4 Modeling & Procedure

4.1 What to Model!

We need to model both documents and queries into numbers so that we can get a score for each pair of document and query. These scores can then be compared using any similarity metric to get the most similar pairs of documents and queries. In this project, these numbers are generated using TF-IDF modeling technique.

4.2 TF-IDF Model

TF-IDF is short for term frequency – inverse document frequency. It is a numerical metric that represents how important a word is to a document in a collection of documents[5]. It is used as a weighting factor in query searches in the field of information retrieval. It is used by different similarity ranking metrics explained in the following subsections.

4.2.1 Cosine Similarity Metric

It represents each document and query as a vector. Similarity of query to a document is represented by the correlation between their vectors. One metric of this correlation is the cosine of the angle between their vectors. It is better used when comparing elements with the same nature like documents with each other[3] $cos(\vec{d_1}, \vec{d_2})$.

Equation for Cosine Similarity is represented as follows:-

$$cos(\vec{q}, \vec{d}) = \frac{q.d}{|\vec{q}||\vec{d}|} \tag{1}$$

- $\vec{d} = (x1, x2, x3, ..., x_n)$ is a vector in an n-dimensional vector space. Each x corresponds to a term in the document representing TF-IDF value of that term.
  While $|\vec{d}| = \sqrt{\sum_{i=1}^{n} x_i^2}$

- $q = (y1, y2, y3, ..., y_m)$ is a vector in an m-dimensional vector space. Each y corresponds to a term in the query representing TF-IDF value of that term.
  While $|\vec{q}| = \sqrt{\sum_{i=1}^{n} y_i^2}$

- $q.d$ is the dot product of the two vectors. Terms present in the query vector $\vec{q}$ and not in in document vector $\vec{d}$ are considered having zero values in $\vec{d}$.
  So it can be evaluated as $q.d = \sum_{i=1}^{m} x_i * y_i$ only if x_i is the same as y_i

4.2.2 BM25 Metric

Okapi BM25 is a ranking function used by search engines to estimate the relevance of documents to a given search query[4]. It represents state-of-the-art TF-IDF-like retrieval functions used in document retrieval. It ranks a set of documents based on the query terms appearing in each document, regardless of their proximity with the document.

Equation for BM25 is represented as follows:-

$$\text{score}(D, Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{\text{avgdl}})} \tag{2}$$

$$\text{IDF}(q_i) = \ln(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1) \tag{3}$$

where $f(q_i, D)$ is q_i's term frequency in the document D, $n(q_i)$ is the number of documents q_i appeared in, $|D|$ is the length of the document D in words, and avgdl is the average document length in the text collection from which documents are drawn. k_1 and b are free parameters which can be tuned. The default values of b = 0.75 and k1 = 1.2 work pretty well for most corpora[2].

Here it's used improve over the performance of the baseline model using cosine similarity ranking function by doing the following steps:-

- First, rank and retrieve the top 1000 documents for a query with the baseline model.
- Second, use bm25 ranking function to re-rank these 1000 documents to retrieve the top 50 documents.
- Third, use bm25 ranking function to rank the sentences in these top 50 documents, and retrieve the top 50 sentences.

5 Evaluation & Results

5.1 Baseline Document Retrieval Model

Table 1 shows the average precision resulted from using Cosine Similarity as a metric. There are two approaches taken for the cosine similarity equation as follows:-

- Take the denominator of the cosine similarity equation into consideration: This resulted into a very low mean average precision later, this maybe attributed to the fact that dividing by (length of the query * length of the document) results into very small numbers especially for large documents even if it contains the answer and is relevant. While small documents that have some of the same terms as the query, but don't have the answer and not relevant, it results into a much higher score although this shouldn't be correct.

- Ignore the denominator of the cosine similarity equation: This resulted into a higher average precision later, this can be attributed to the fact that taking only the nominator will encourage large and relevant documents that has most of the query terms including the answer to have very high scores, since we don't normalize the score by its length. In this problem, it increased the probability of getting relevant documents and the mean average precision over the previous approach.

This is verified later by observing the relevant documents using (patterns) data for query number 1, and found out that the most relevant document for this query is indeed larger, while there are other documents which aren't relevant but has the same query terms, and these are wrongly favoured by cosine similarity formal equation.

The second approach is taken as it resulted into higher mean average precision, which will benefit later the more advanced model in achieving better average precision.

An example of the top document scored for query number 1.

****** Query No. 1 ******

[’who’, ’is’, ’the’, ’author’, ’of’, ’the’, ’book’, ’the’, ’iron’, ’lady’, ’a’, ’biography’, ’of’, ’margaret’, ’thatcher’]

****** Highest scored document ’FT921-1445’ ******

[’the’, ’economist’, ’known’, ’as’, ’the’, ’father’, ’of’, ’monetarism’, ’and’, ’a’, ’guru’, ’of’, ’former’, ’prime’, ’minister’, ’margaret’, ’thatcher’, ’professor’, ’friedrick’, ’von’, ’hayek’, ’died’, ’at’, ’his’, ’home’, ’in’, ’freiburg’, ’in’, ’breisgau’, ’germany’, ’aged’, ’92’]

We observe that the highest score document has indeed some of the same terms as the query, but it doesn’t have the required answer. The answer should be **”Hugo Young”**.

We also observe that the highest score document for query number 1 doesn’t exist in the relevant documents retrieved using *Patterns data*, with small length of *54988.499*.

This maybe an indication for the point mentioned before about the cosine similarity equation. Relevant documents that have the answer are longer than non-relevant documents that don’t have the answer but still score high since they have overlapping query terms.

5.2 Advanced Document Retriever with Re-Ranking

Table 2 shows the average precision resulted from using bm25 as a re-ranking metric after ranking with the baseline model.

An example of the top document scored for query number 1.

****** Query No. 1 ******

[’who’, ’is’, ’the’, ’author’, ’of’, ’the’, ’book’, ’the’, ’iron’, ’lady’, ’a’, ’biography’, ’of’, ’margaret’, ’thatcher’]

****** Highest scored document ’LA111289-0002’ ******

[..., ’as’, ’a’, ’militant’, ’puritan’, ’in’, ’this’, ’same’, ’revisionist’, ’mold’, ***’hugo’, ’young’****, ’the’, ’distinguished’, ’british’, ’journalist’, ’has’, ’performed’, ’a’, ’brilliant’, ’dissection’, ’of’, ’the’, ’notion’, ’of’, ’thatcher’, ’as’, ’a’, ’conservative’, ’icon’, ’in’, ’the’, ’iron’, ’lady’, ’young’, ’traces’, ’the’, ’winding’, ...]*

We observe that the highest score document has indeed some of the same terms as the query, but also has the answer to the required query.

We also observe that the highest score document for query number 1 is indeed exists in the relevant documents retrieved using *Patterns data*, with large length of *1282454.277*.

Ranking Metric	Equation	Average Precision
Cosine Similarity	$cos(\vec{q}, \vec{d}) = \frac{q.d}{\lvert\vec{q}\rvert\lvert\vec{d}\rvert}$	0.008
Cosine Similarity (without denominator)	$cos(\vec{q}, \vec{d}) = q.d$	0.164

Table 1: Baseline Document Retrieval Model Average Precision

Ranking Metric	Average Precision
BM25 Score	0.462

Table 2: Advanced Document Retriever with Re-Ranking Average Precision

5.3 Sentence Ranker

Table 3 shows the mean reciprocal rank resulted from using bm25 as a ranking metric for ranking sentences in the top 50 documents retrieved from the previous step (Advanced Document Retriever with Re-Ranking).

****** Query No. 1 ******

['who', 'is', 'the', 'author', 'of', 'the', 'book', 'the', 'iron', 'lady', 'a', 'biography', 'of', 'margaret', 'thatcher']

****** Highest scored sentence ******

*['the', 'iron', 'lady', 'a', 'biography', 'of', 'margaret', 'thatcher', 'by', **'hugo', 'young'**, 'farrar', 'straus', 'giroux', 'the', 'central', 'riddle', 'revealed', 'here', 'is', 'why', 'as', 'a', 'woman', 'in', 'a', 'mans', 'world', 'margaret', 'thatcher', 'evinces', 'such', 'an', 'exclusionary', 'attitude', 'toward', 'women']*

We observe that the highest score sentence has indeed some of the same terms as the query, and it has also the answer to the required query.

We observe that the highest score sentence for query number 1 exists in the relevant sentences retrieved using *Patterns data*

Retrieved sentences are only compared with relevant sentences retrieved only from the top 50 documents, and not with all the sentences from all the documents in the corpus.

Ranking Metric	Mean Reciprocal Rank
BM25 Score	0.393

Table 3: Sentence Ranker Mean Reciprocal Rank

References

[1] Introduction to information retrieval. https://nlp.stanford.edu/IR-book/information-retrieval-book.html.

[2] Elastic blog. “practical bm25 - part 3: Considerations for picking b and k1 in elasticsearch. https://www.elastic.co/blog/practical-bm25-part-3-considerations-for-picking-b-and-k1-in-elasticsearch, 2018.

[3] Wikipedia contributors. Cosine similarity — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Cosine_similarity&oldid=972200804, 2020.

[4] Wikipedia contributors. Okapi bm25 — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Okapi_BM25&oldid=981003569, 2020.

[5] Wikipedia contributors. Tf–idf — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Tf%E2%80%93idf&oldid=979969361, 2020.