Information Retrieval and Web Search

Lecture 07

Lecture by Dr. Inbal Budowski-Tal
Typeset by Steven Karas

2018-05-16
Last edited 20:54:08 2018-05-16

Disclaimer These lecture notes are based on the lecture for the course Information Retrieval and Web Search, taught by Dr. Inbal Budowski-Tal at IDC Herzliyah in the spring semester of 2017/2018. Sections may be based on the lecture slides written by Dr. Inbal Budowski-Tal.

Agenda

- Scoring in a complete system

1 Recap

1.1 tf-idf

Last week, we introduced the TF-IDF ranking scheme. We denote the log weighted term frequency of a term t in a document d as:

$$w_{t,d} = \begin{cases} 1 = \log_{10} tf_{t,d} & \text{if } tf_{t,d} > 0 \\ 0 & \text{else} \end{cases}$$

The document frequency df_t is the number of documents that a term t appears in. We denote the idf weight, which is a measure of the informativeness of a term t as:

$$idf_t = \log_{10} \frac{N}{df_t}$$

The tf-idf is the product of the weighted term frequency and the idf weight.

1.2 cosine similarity

Because tf-idf provides us with vectors for terms, we want a useful measure of the similarity between terms. As such, we use cosine as a distance measure:

$$\cos(\vec{q}, \vec{d}) = \text{SIM}(\vec{q}, \vec{d}) = \frac{\vec{q}}{|\vec{q}|} \cdot \frac{\vec{d}}{|\vec{d}|} = \sum_{i=1}^{|V|} \frac{q_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}} \cdot \frac{d_i}{\sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

1.3 Weighted example

A fully worked example using the LNC.LTN scheme can be found on slide 9.

2 Ranking

Motivation Unranked retrieval can be extremely difficult to craft precise queries, especially as it's unreasonable to examine hundreds or thousands of results. Ranking effectively reduces a large result set to a very small one. Empirical research done by Dan Russell at Google showed through eye tracking and recording video of users searching for various terms in a controlled environment that the overwhelming majority of users only look at the first 3 results, and open only the first. To control for presentation bias, they A/B tested swapping the order of the first two results. This showed that presentation bias has a huge effect.

Problem for cosine similarity If an extremely rare term is used in a query, next to an extremely common word, then it may overwhelm the other query terms. To compensate for that, we can squash the growth of teh term frequency:

$$\text{corrected } tf = \frac{tf}{tf + K}$$

Where K is selected to control the aggressiveness of this squashing.

An alternative approach also compensates for the relative information imparted by repititions in longer documents:

$$\text{corrected } tf = \frac{tf}{tf + \frac{K \cdot |d|}{\text{avg}(|d|)}}$$

Storing term frequencies We need to store the $tf_{t,d}$ in addition to the document id for each posting in the index. In practice, it's better to store the term frequency in integer form rather than as a log frequency for compression purposes. Unary encoding is useful for storing term frequencies, but requires a bit per term in each document plus a bit per unique term. However, bitwise compression can be very useful.

2.1 K-selection

The problem of selecting the k largest items from a collection is a well studied problem, with various algorithms with differing tradeoffs. A common approach for IR is to use a min-heap and run `heappoppush` for each document. This gives a $O(N \log k)$-time solution. If we can preprocess the distances between documents, we can construct data structures that answer these queries in sublinear time.

Heuristics In practical terms, it's often useful to prune the search space. If we store the postings in a query-independent ranking, then we can compute the composite score, and terminate early.

2.1.1 PageRank

PageRank was the PhD dissertation of Larry Page and Sergey Brin, who founded Google. The gist is to consider the crawled documents as a markovian chain, and efficiently computing the eigenvectors of the adjacency matrix. By sorting the documents by pagerank, we can terminate queries early when the composite score cannot exceed the document ranking. However, this requires normalization for both the document rankings and the query distance metric.

3 Next Session

I won't be able to attend the lectures of this course for the remainder of the semester. Good luck!

References

[1] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.