Relevance Weighting of Search Terms

Paper:

- https://www.staff.city.ac.uk/~sbrp622/papers/RSJ76.pdf

Motivation for relevance weighting of search terms

Like why don't you just look for an exact match? That would be Boolean Retrieval.
Two limitations:

1. If isn't a perfect match to query, we don't get any documents
2. If there are too many matches, some should match better than the others
   What we really want is a Ranked Retrieval, where multiple documents are relevant to a given query, and we also want to rank them. This paper proposes weighting terms to distinguish documents that all match at least some query terms, so that more relevant ones are ranked higher.

Question

Is there a connection to Page Rank for google search? No, that's based on a page's popularity, though in practice it could be used in compibnation.

Notes / Terminology

Term vs query:

- A *term* refers to a specific keyword or indexing unit found within document (ex: "relevance")
- A *query* refers to the request for information submitted by a use ("find papers on relevance weighting"), it's just a set of terms

Relevant document = document relevant to query.

Key terminology:

- N is total number of documents
- n is total number of documents indexed by a *term*
- R is total number of *relevant* documents across all documents
- r is total number of *relevant* documents that contain the *term*

We can derive these definitions:

- $n - r$ is the number of *non-relevant* documents that contain the *term*
- $N - R$ is the total number of *non-relevant* documents
- $N - n - R + r$ is the number of *non-relevant* documents that do not contain the *term*

Relevance weighting: we weight each term on how relevant they are to the query.

Why not just $w = \frac{r}{R}$?

- This equation doesn't account for occurrences of term in non-relevant documents. Imagine a term is in all documents
- So we want probability that captures how often we see this term in non-relevant documents ($n - r$) over all non-relevant documents ($N - R$).

This is actually what the paper talks about: F1, F2, F3, F4, which are different equations.

Statistical Terminology:

- I1: distribution of terms in relevant document is independent and distribution of term in all documents in independent
- I2: Distirbution of terms in relevant document is indepenent and distribution of term in NON-RELEVANT documents is indepdent
- O1: Relevance of term is based only on the presence of the term in the document
- O2: Relevance of term is based BOTH on presence and their absence from non-relevant document

Retrieval vs. ranking

I was getting confused about this paper because I was confused about the fact that they are using r to predict how relevant a document is, but and comparing it against a ground truth relevance. But r is already sort of a ground truth..?

But no, it's about ranking the documents. And in the Cranfield dataset, we want to make sure the matched documents for a query are consistent from the top ranked documents.

One would think about dependence on presence implies dependence on absence. But the authors show that explicitly computing it makes a difference.

The following equations are derived from the contingence table:

Simple term match

$$F0 : w = -\log \frac{n}{N}$$

$$F1 : w = \log \frac{\frac{r}{R}}{\frac{n}{N}}$$

$$F2 : w = \log \frac{\frac{r}{R}}{\frac{n-r}{N-R}}$$

- relevant documents containing term over non-relevant documents containing term

$$F3 : w = \log \frac{\frac{r}{R-r}}{\frac{n}{N-n}}$$

$$F4 : w = \log \frac{\frac{r}{R-r}}{\frac{n-r}{N-n-R-r}}$$

F1 and F3 vs. F2 and F4

$F1$ and $F3$ compute ratio of relevant documents to total number of documents (O1 principle)

- F2 and F4 compute ratio of relevant to non-relevant documents (O2 principle)

Document Ranking

Once we can calculate a weight for each term, we can use it to compute a matching coefficient, specifically a sum of the weights of each term:

$$Score(d, q) = \sum_{t_i \in q \cap d} w_t$$

- d is the document
- q is the query

- w_t is the relevance weight of term t, computed with F1 / F2 / F3 / F4

Experimental Setup

They use Cranfield, which contains queries and list of relevant documents, and contains a ranking. They compare the computed values against the human relevance judgments (treated as ground truth), and plot the precision recall curve.

- Precision: Of the documents you retrieved, what fraction are relevant?
- Recall: Of all the relevant documents that exist, what fraction did you retrieve?

The paper talks about two kinds of experiments

1. Use of weights *retrospectively*, where we already know r and R, and can assess how correct a particular method is with simple proportion estimates
2. Use of weights *predictively*, in which we use an estimate of r and R on new unknown queries, using the average of the w_t terms

Questions

What are the authors trying to do? Articulate their objectives.
The authors are present a theoretical and practical analysis of relevance weighting in information retrieval, i.e. computing a weight for different terms of a query, and show that a particular formula (F2,F4) exploits information about relevant and non-relevant documents more effectively. They also illustrate an application of these techniques through real examples and prove that F4 perfroms better in the Cranfield dataset.

How was it done prior to their work, and what were the limits of current practice?
Prior to this work, search term weighting was already an established practice, but the methods to score these terms were much simpler (F0). These formulas failed to explicitly exploit available information about document relevance to the query - specifically, they rewarded presence of terms in relevant documents, but didn't penalize presence of those terms in non-relevant documents.

What is new in their approach, and why do they think it will be successful?
They propose a systematic theoretical framework for relevance weighting. They derive $F1, F2, F3, F4$ formulas from a contingency table and show that F2 and F4 explicitly recognize term absence in calculating probable relevance (via $n - r$), embodying ordering Principle O2. They show that the O1 assumption is incorrect, and that O2 is much more effective and provide a theoretical analysis.

What are the mid-term and final "exams" to check for success? (i.e., How is the method evaluated?)
They empirically check using the Cranfield 1400 collection and the Cranfield 800 odd-numbered sub-collection. This evaluation involves computing precision and recall metrics, and comparatively comparing retrieval performance.

Limitations:
In this paper, an individual term has the same weight in different documents under the same query. This means if that multiple documents has the same terms, even though those terms appear more frequently, the document will not get ranked higher. Something like TF-IDF gets around that