Information Retrieval Lecture - Relevance Feedback

Colm O'Riordan

October 2020

1 Introduction

Relevance feedback has proven to be highly effective in improving information retrieval performance. Upon receiving returned articles, the user may provide relevance judgments for these articles. These relevance judgments may subsequently be used to guide the matching function for the retrieval system.

Typically, on presentation of the results from the filtering system, the user is asked to identify which documents are relevant and which are non-relevant. This information, together with the current user query Q_k, is then used to form a new query Q_{k+1}.

The feedback mechanism adopted is clearly dependent on the representation (and comparison) strategies in use, but generally involves adding or indeed removing terms to the query in addition to adjusting any of the weights of the existing terms. For any of the models adopted (Boolean, vector space, etc.), feedback techniques can be developed.

Numerous techniques have been adopted to obtain user feedback. Explicit feedback mechanisms require the user to offer feedback on the returned set; implicit mechanisms involve attaining evidence from the user's behaviour to alter the query.

Query expansion and modification can also be carried out in a manual manner as typified in many user's interactions with web search engines where the user, upon viewing the returned set, modifies the query in the hope of improving the quality of the returned set. Most modern information retrieval systems attempt to perform this query modification in an automatic manner.

Despite the advances in the field of information retrieval, many current systems still have limited recall; few relevant documents will often be returned in response to user specified queries and rarely are all the relevant documents returned. Harman [1] disusses the limits of providing increasingly better results based solely on the initial query, and suggests modification to that query to further increase performance.

Given the initial search made by the system upon receiving a user query, the user is a given a number of documents. The user indicates which of the retrieved documents are useful (relevant). The system then automatically reformulates the original query based upon these user relevance judgments. The process iterates until the user's information need is satisfied.

In addition to using just user-feedback, useful evidence can be gleaned from the document collection (or subsets of the document collection). Oftentimes, analysis of the returned set is used to expand or modify the query. This is termed *local analysis* of *pseudo-feedback*.

In summary, the process of relevance feedback in information retrieval systems involves modification of the user query to improve the relevance of the result. Query modification can occur in a manual or an automated manner. The latter is of more interest to us in this paper and will be type considered in the remainder of the paper. Furthermore, user feedback can occur in an explicit or implicit manner. Another distinction made is that query modification can be based on user-feedback or analysis of the returned set (or the whole set).

2 Relevance Feedback for the Vector Space model

In the vector space model, we represent both documents and queries as a weighted vector. Comparison involves measuring the cosine the angle between the vectors.

In the 1960s, Rocchio [3] demonstrated positive results in experiments in query modification in the vector space model. He showed how the query can be created through vector addition and subtraction, is the sets of relevant and non-relevant documents are known.

Weights of query terms occurring in the relevant documents are increased; those in non-relevant documents are reduced. Queries can be automatically expanded by adding terms, not in the original query, that occur in the relevant documents.

The Rocchio method is based on the observation (and assumption) that relevant documents have similarly weighted term vectors.

Let D_r denote the set of relevant documents returned by the system, D_n the set of non-relevant documents returned by the system and C_r be the relevant documents in the whole collection.

Let's assume C_r is known for some query q. The best possible query to obtain the set C_r as the returned set can be generated as:

$$\vec{q} = \frac{1}{|C_r|} \sum_{d_j \in C_r} d_j - \frac{1}{N - |C_r|} \sum_{d_j \notin C_r} d_j$$

As the set C_r is not known in advance, the best we can do is estimate C_r. The retrieval system returns a set D_r of relevant documents. The user will in an ideal system identify D_r for the system (or at least a representative subset). Thus, a new query can be calculated which will hopefully give better performance against the document set:

$$\vec{q} = \frac{1}{|D_r|} \sum_{d_j \in D_r} d_j - \frac{1}{N - |D_r|} \sum_{d_j \notin D_r} d_j$$

The Rocchio method involves modifying the above equation slightly to give:

$$\vec{q} = \alpha \vec{q} + \frac{\beta}{|D_r|} \sum_{d_j in D_r} d_j - \frac{\gamma}{N - |D_r|} \sum_{d_j \notin D_r} d_j$$

where α, β and γ are constants that are used to determine the importance of feedback and the relative importance of positive feedback over negative feedback.

Variations on the Rocchio method exist and include:

1. Ide-Regular [2]:

$$\vec{q} = \alpha \vec{q} + \beta \sum_{d_j \in D_r} d_j - \gamma \sum_{d_j \in D_n} d_j$$

2. Ide Dec Hi which uses only the highest ranked non relevant document as the only document used in negative feedback.

$$\vec{q} = \alpha \vec{q} + \beta \sum_{d_j \in D_r} d_j - \gamma MaxNR(d_j)$$

where $MaxNR(d_j)$ is the highest ranked non-relevant document.

The methods above can be used when we have vector based representations of the documents.

3 Document Set Analysis

It is generally accepted that in many domains most users typically use very short queries and are not likely to take the trouble to construct long and carefully stated queries. Such short queries may lack many useful words that, if provided by the user, can be very effective search terms. Automatic query expansion via relevance feedback is an effective technique commonly used for altering queries to provide more evidence to the system and hence obtain more accurate results.

Casual users seldom provide a system with relevance judgement needed to perform effective relevance feedback. This further motivates the need to develop techniques that automatically expand queries.

3.1 Local Analysis

In local analysis, documents which are retrieved are examined at query time to determine terms for query expansion. Most systems typically involve the development of some form of term-term correlation matrix to quantify the connection between two terms.

This correlation can be calculated over the whole document set or in the context of the current query. The current query is then expanded by including terms correlated to the query terms.

Different means are available to develop the correlations.

3.1.1 Association Clusters

A correlation matrix can be calculated such that each cell in the matrix $M_{i,j}$ indicates the correlation between terms i and j. One approach to calculating this is to consider the amount of times the terms co-occur in the document collection:

$$M_{i,j} = \frac{freq_{i,j}}{freq_i + freq_j - freq_{i,j}}$$

Given these calculations, we can develop an association cluster for each term t_i. We select the i^{th} row from the matrix and select the top N values from the row; these values correspond to the top N correlates for the term t_i.

For any query q, select a cluster for each query term to create $|q|$ clusters.

N is usually kept quite small to prevent the query from becoming too large and potentially 'drifting' in terms to topic captured in the query. All the terms from the $|q|$ clusters may be taken but choosing the values with the highest summed correlation.

3.1.2 Metric Clusters

The main flaw with association clusters lies in their inability to take into account terms' positions within a document. The degree of correlation between two terms is calculated solely on their co-occurrence within a document and not whether they occur close to each other.

Metric clusters represent an attempt upon this by taking into account the distance between two terms when calculating correlations.

Let $dis(t_i, t_j)$ be the distance between two terms t_i and t_j occurring in the same document. If t_i and t_j are in different documents, then $dis(t_i, t_j) = \infty$.

We can define the weights in our term-term correlation matrix as:

$$M_{i,j} = \sum_{t_i, t_j \in D_i} \frac{1}{dis(t_i, t_j)}$$

We can define clusters for each query term as before.

3.2 Scalar Clusters

Scalar clusters represent an approach which tries to improve upon the previous approaches by attempting to pay attention to the context in which a term occurs. The neighbourhood of terms surrounding a term is considered when calculating the correlation between these terms. It is based on the heuristic that if two terms have similar neighbourhoods then there is a high correlation between terms.

The size of the neighbourhood to be considered can be taken as a fixed sized set of terms either side of the term in the question or can be taken as the current sentence of paragraph.

Similarity can be based on comparing the two vectors representing the neighbourhoods using the cosine similarity measure. This measure can be used to define term-term correlation matrix and the procedure continues as before.

4 Global Analysis

Global analysis involves performing an analysis on the whole document collection and not just the returned set. The degree of correlation between two terms is calculated based on their occurrences across the whole collection.

A term-document matrix is created (note, in many IR systems this will be created to facilitate querying). We can view this matrix as indexing each term by the documents in which they occur, as opposed to the usual interpretation of it being each document indexed by terms which it contains.

The weights assigned to each term-document value is based on the term weight assigned by our weighting scheme.

It is then possible to calculate similarity between any two terms by taking some measure of similarity between their two vectors (cosine measure). We can use this to expand our queries.

There are related approaches based on clustering of documents and on word embeddings which we will discuss in a later lectures.

5 Other Issues

Similar approaches can be applied in the domain of information filtering. In the IF domain, a user's query tends to exist for a much longer period of time; hence, the approach taken do not change the query or profile as quickly.

Learning mechanisms and feedback mechanisms are implicit in many other IR models - neural and probabilistic approaches incorporate the notion of feedback in their models.

There are many issues and problems related to obtaining use feedback; this is due to a number of reasons:

- Users tend not to give a high degree of feedback
- Users are typically inconsistent with their feedback
- Explicit user feedback does not have to be strictly binary (a range of values may be more appropriate)

Implicit feedback can also be used in an effort to compensate for the lack of explicit user feedback. One can make certain assumptions that a user finds an article based on certain heuristics such as:

- a user reads an article
- user spends a certain amount of time reading an article
- user saves or prints an article

These implicit measures are rarely as trustworthy as explicit feedback.

References

[1] Donna Harman. Relevance feedback revisited. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 1–10, 1992.

[2] Eleanor Ide. New experiments in relevance feedback. *The SMART retrieval system: Experiments in automatic document processing*, pages 337–354, 1971.

[3] Joseph Rocchio. Relevance feedback in information retrieval. *The Smart retrieval system-experiments in automatic document processing*, pages 313–323, 1971.