Information Retrieval - Lecture 2

Colm O'Riordan

September 2020

1 Introduction

The field of information retrieval can be seen as a natural progression from data retrieval. It attempts to develop models and systems suitable for the vast quantity of unstructured information available today.

Information Retrieval differs from traditional data retrieval in a few ways. Data collections tend to be well structured collections of related items in which each item is usually atomic in nature with a well defined interpretation. Information, on the other hand, is usually semi-structured or unstructured.

Data retrieval involves the selection of a fixed set of data based on a well-defined query (usually expressed in SQL or equivalent). Information Retrieval (IR) involves the retrieval of documents of natural language which is typically not structured and may be semantically ambiguous.

An IR system must make and effort to 'interpret' the semantic content of the document and rank the documents in relation to users' information needs.

2 Information Filtering

The term *information filtering* is often also used to describe systems that identify relevant information for users in response to an information need. The fields of information retrieval and information filtering have a lot in common - similar representations, mathematical models and comparison approaches are involved. They differ in two main regards: the nature of the information need and the nature of the document collection. In IR, we typically deal with one-off information needs and a relatively static collection. Information filtering, in contrast, considers the information need as being a long-standing information and the document collection is viewed more as a 'stream' of information with the system making a decision regarding its relevance without access to the complete collection.

3 User Role

In traditional information retrieval and library systems, the user role was pretty well defined and understood in that the user formulated a query, viewed the results and possibly offered feedback to the system. The collections tended to be centralised and the user actions limited. In more modern IR systems, with the increased popularity of the hypertext paradigm, the user usually intersperses querying with browsing links. In doing so, the user's information need often evolves and changes during the interaction. The more complex interaction model poses some interesting challenges for IR researchers.

4 IR System Architecture

At a high level, we can view an IR system as comprising:

- Document Set/Collection (e.g web, articles, tweets, etc.)
- Queries (representations of users' information needs)
- Pre-processing components (e.g stemming)
- Comparison algorithm (some means to determine the similarity between the query and the document)
- User Feedback module

In the most general case, the documents in the document collection and the information query are in the same format - namely, natural language. These are then processed in some manner to produce an internal representation of both document and query. This representation is also a format suitable for the comparison algorithm. This representation and comparison algorithm will depend on the underlying model chosen in the IR system.

The comparison algorithm should return some estimate of relevance (or similarity) for each document-query pair, i.e., an estimate of how likely the document is to satisfy the information need expressed in the query. These estimates usually allow the system to rank all documents with respect to relevance to a query. The top-ranked documents can then be presented to the user.

The final component is the feedback module; here the user can offer feedback on the usefulness/relevance of the returned documents. Evidence from this feedback is then used to typically modify the user's query. This can include the incorporation of new query terms; the removal of existing term and/or the possible re-weighting of terms. The goal is to create a better query for the user's information need leading to a better return set being returned.

5 Pre-processing overview

The document pre-processing phase involves applying a well known set of techniques to the document collection to convert it to a format more suitable to the task at hand. There are many potential pre-processing approaches with certain approaches developed for specific requirements of particular domains or particular user needs. The include, among others:

- **Stemming:** Stemming algorithms attempt to remove common suffixes from terms occurring in the documents. The overall goal is to reduce similar words to a common root form by identifying morphological derivations of words. There are many approaches in the literature and in commercial systems. Lovin's algorithm and Porter's stemmer are two of the well-known classic stemmers.
- **Stop word removal:** This involves the removal of highly frequent words/terms from documents. These words add little semantic meaning to the document and usually include articles and conjunctions.

6 IR models

There have been many information retrieval models proposed in the literature. These can be broadly categorised as: Boolean models, Vector based models and probabilistic models. A model can be viewed as a tuple:
$[D, Q, F, R(q_i, d_j)]$
where:

- D is the set of logical representation of the documents
- Q is the set of logical representations of the user information needs
- F is the framework adopted for modelling the representations and their relationships
- R is a ranking function which produces a ranking of documents with respect to estimated relevance to a query q_i

In most models, we have a set of index terms $t_1 \dots t_n$. A weight $w_{i,j}$ is assigned to each term t_i occurring in the document. We can also view the query (in most cases) as a set of terms with associated weights.

7 Boolean Model

The Boolean model is based on the simplistic concept that is a document contains a term or a set of terms that satisfy the query (a Boolean expression), then the document is relevant to that query. It is based on the set theory and Boolean algebra. It is a simple model and often used. The advantages of the model (simplicity, clean formalism) are often overshadowed by its limitations:

- people often have difficulty formulating expressions due to subtle differences in the natural language usage or "OR" and "AND" and the Boolean algebraic meaning
- it suffers badly from natural language features such as synonymy and polysemy.
- it pays no attention to frequency of terms in documents.
- documents are considered either relevant or non-relevant; there is no potential for partial matching; no real ranking allowed.
- terms in documents are consider independent of each other.

8 Vector Space Model

The vector space model attempts to improve upon the Boolean model by removing the limitation of binary weights for index terms. In the vector-space model, terms can have a non-binary value for both queries and documents.

There terms weights are used in the comparison between documents and queries. We can the sort the documents based on their degree of similarity and return a ranked list to the user.

Each term in both the documents and the queries will have an associated weight. Hence we can represent documents and queries as n-dimensional vectors:

$$\vec{d_j} = (w_{1,j}, w_{2,j}, \dots w_{n,j})$$

$$\vec{q} = (w_{1,q}, w_{2,q}, \dots w_{n,q})$$

We can calculate the similarity between a document and a query by calculating the similarity between the vector representations. We can measure this similarity by calculating the cosine of the angle between the two vectors.

$$sim(d_j, q) = \frac{\vec{d_j} \bullet \vec{q}}{|\vec{d_j}||\vec{q}|}$$

$$\Rightarrow sim(d_j, q) = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,q}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\sqrt{\sum_{i=1}^{n} w_{i,q}^2}}$$

This provides a neat formalism for calculating a measure of similarity between a document and a query.

9 Weighting Scheme

In the above equations, the similarity is a function of the number of co-occurring terms between a query and a document; and of the weights of the terms in document in the query.

Clearly the assignment of weights to terms is of importance.

Early work (Salton and Fox) proposed two main heuristics:

- **tf**: term frequency; we can view a term's frequency in a document as a means of quantifying how well a term describes a document. The more frequent a term occurs in a document, the better it is at describing that document and vice versa.
- **idf** inverse document frequency; another heuristic that can be applied is based on the observation that a term that occurs frequently across all documents does little to distinguish one document from another.

These heuristics give rise to a family of weighting schemes known as **tf-idf** weighting schemes. One early version was provided by Salton and Fox (1983).

$$w_ij = f_{i,j} \times log(\frac{N}{n_j})$$

where:

- $f_{i,j}$ is some function of the frequency of t_i in document d_j
- N is the number of documents in the collection
- n_i is the number of documents in the collection that contain term t_i