Information Retrieval 24-25

Notes

University of Pisa

M.Sc. in Computer Science

Contents

Chapter 1

Introduction

Information retrieval is the process of finding relevant material of unstructured nature from large collections. The "material" is usually documents, web pages, or multimedia content. Originally, information retrieval was something only a few professionals interacted with. Nowadays, hundreds of millions of people engage with information retrieval systems when they, for example, use a web search engine or search through their email.

The two key aspects considered when evaluating the quality of an IR system are **effectiveness** and **efficiency**. The first refers to the capability of the system to produce a satisfactory result; the second refers to how quickly it does so. To guarantee a certain level of quality, some operations are done offline, such as document indexing, feature processing (e.g., term frequency, metadata), training of a learn-to-rank model to produce the order in which documents will be shown to the user, and so on. Still, many operations must be done on-line, such as query expansion and processing, index and feature lookup, and usage of the ranking funcion. If we consider the example of a search engine (SE), we are used to get back a response in a very short amount of time, despite the fact that in order to find the collection of documents presented to us, a lot of different operations must have been performed (i.e., the system must be efficient). Additionally, we also expect that those documents are the most relevant ones found in the collection, and that they are presented in the order of relevancy (the system must be effective). If those two ideas do not hold, we're unlikely to actually use the system for an extended amount of time.

The following chapters will go in detail about the different components of IR systems, and each will focus on how effectiveness and efficiency can be guaranteed. The key aspects that will be considered are:

- **Language properties:** how does language influence retrieval? What does it mean to retrieve a piece of text? How are documents scored and presented?

- **Auxiliary data structures:** e.g., inverted indexes;
- **Query processing:** how are queries expanded from the form provided by the user into one which can be "read" by the system?
- **Data storage and compression:** how can data be compressed efficiently?
- **Learning-to-rank models:** how is machine learning used in an IR system to produce a ranking (based on available ranked data)?
- **Neural IR:** how can Deep Learning and specifically Large Language Models be used in IR?

Chapter 2

Evaluation

To evaluate the quality of a IR system, say a SE, we may ask questions such as: how fast does it index a collection, how fast does it search (efficiency)? Or, does it recommend good related pages/products to buy to the user (effectiveness)? However, these questions alone do not provide any objective information about the intrinsic quality of the SE. We could say that a SE is "good" if it makes its users happy; but to measure this happiness, we can use different definitions: for example, how many times a search result is clicked, how long users stay on the same webpage, how often they return to use the SE. Since happiness by itself is impossible to measure, a commonly used proxy is **relevance** of search results.

2.1 Relevance

In order to measure relevance, three elements are needed:

- A benchmark document collection;
- A benchmark suite of queries;
- An assessment of either **relevant** or **non-relevant** for each query and document.

To construct the benchmark, we would have to analyze each possible pair of query and document and assign a relevance to it. Relevance assessment can be binary (relevant/not relevant), or multi-valued (0, 1, 2, 3 ...) for more nuance. Obviously, since assigning a relevance value to each query-document pair in a collection is way too expensive, a subset of the documents is used instead.

Assigning relevance must be done externally by humans. Some companies use crowdsourcing platforms (e.g., Amazon Mechanical Turk) to present pairs to low cost, not

highly qualified workers. This solution is cheap, but the outcome may not be as good as one produced by professionals.

But how are the queries defined? They must be relevant and suitable to the documents in the collection, and must be representative of user needs (i.e., they should resemble a normal query done by a person). One way to find them is to sample directly from existing query logs of the SE, if available. For classical, non-Web IR systems, these query logs may be nearly empty, as the query rate tends to be slow. In this case, experts may handcraft "user needs" and associated queries. Among popular public test collections is **TREC** (**Text REtrieval Conference**), where focus areas are called **tracks**; each track has a motivating use case, usually an abstraction of a user task. In practice, TREC consists of:

- A set of documents;
- A set of information needs (called **topics**);
- Relevance judgements that indicate which documents should be retrieved by which topics.

The result of a retrieval system executing a task on a test collection is called **run**. The technique first used to select the sample of documents to present to a human judge is **pooling**: the top results for a set of runs are combined to form a pool and only those documents are judged. Since this method automatically assumes that all unpooled documents are not relevant (so they remain unjudged), alternative methods have been investigated by TREC tracks to obtain judgements that support fair evaluation.

Note that TREC does not contain any query, and only generic user needs. Participants are free to define (manually or automatically) actual queries for their specific IR system.

2.1.1 Measures

Evaluation of Unranked Retrieval Sets

Precision and **recall** are binary assessments commonly used to evaluate the effectiveness of an IR system. They are defined on the basis of a set of counts, described by the table below.

	Retrieved	Not retrieved
Relevant	TP	FN
Non-relevant	FP	TN

The two measures are then defined as:

- **Precision:** fraction of retrieved documents that are relevant.

$$Precision = \frac{TP}{TP + FN}$$

- **Recall:** fraction of relevant documents that are retrieved.

$$Recall = \frac{TP}{TP + FP}$$

The **F-Measure** (or **F-Score**) is another metric which condenses both precision and recall; it's calculated as the harmonic mean of the two:

$$F = 2\frac{Prec.Rec.}{Prec. + Rec.}$$

The harmonic mean is alway less or equal than the arithmetic/geometric mean: if the two values are very different, the harmonic mean is closer to their minimum. A weighted variant also exists; let α be the weight assigned to precision and β the weight assigned to recall, such that $\alpha = 1/(1 + \beta^2)$, **weighted F-Measure** is calculated as:

$$F_\beta = (\beta^2 + 1)\frac{Prec.Rec.}{\beta^2 Prec. + Rec.}$$

Evaluation of Ranked Retrieval Results

Precision, recall, and F-score are set-based methods, meaning that they are computed using an unordered set of documents. They can be extended to evaluate the ranked retrieval results returned by search engines. Some of these measures are **Mean Average Precision** (**MAP**), **Precision@K** (**P@K**), **Mean Reciprocal Rank** (**MRR**) for binary relevance, and **Normalized Discounted Cumulative Gain** (**NDCG**) for multiple levels of relevance.

Mean Average Precision Consider a set of documents returned by the SE, ordered by rank. Consider the indexes $K_1, \ldots, K_R$ at which recall increases. Precision is calculated at each K_i, considering only the documents with lower indexes, and the average across all K_i is calculated at the end. MAP is then calculated as the mean of the averages across multiple queries/rankings.

MAP is a macro-averaging measure; each query counts equally, even if only few documents are relevant for certain queries and many are relevant for other queries.

Precision@K MAP takes into account all documents returned by the query. However, especially in Web searches, the number of retrieved documents may be unknown or very high; what truly matters is how any good results are found in the first few pages. To calculate Precision@K, we set a rank threshold K, we compute the number of relevant documents among the top K ranking ones, and calculate P@K as:

$$P@K = \frac{\sum relevant\ documents\ in\ top\text{-}K}{K}$$

i.e., it is the fraction of relevant documents across the top-K retrieved ones. In a similar fashion, we can also calculate **Recall@K** as the fraction of relevant documents that are retrieved among the top-K.

P@K by itself does not average very well over a set of queries, since the number of relevant documents for a query affects the result. Anyway, it (along with **MAP@K**) are largely used for Web searches and recommender systems.

Mean Reciprocal Rank Suppose there is only a single relevant document for a given query. A way to approximate the rank of the correct answer could be to consider the search duration for the user: if he/she takes a long time to find the answer its ranking is low, if instead he/she finds it quickly it is ranked high.

Consider the rank position $rank_i$ of the first relevant document returned by a query $q_i \in Q$. The **Reciprocal Rank** is calculated as:

$$\mathbf{RR} = \frac{1}{rank_i}$$

MRR is the mean of the RRs across multiple queries.

Evaluation of Non-Binary Relevance

A popular measure used to evaluate web searches with non-binary relevance is **cumulative gain**, and in particular **Normalized Discounted Cumulative Gain** (**NDCG**). The idea is that the lower the rank of a relevant document is, the less it is useful for the user, since it is less likely to be visited.

Discounted Cumulative Gain uses graded relevance (or **gain**) as a measure of usefulness; it is accumulated starting at the top of the ranking and may also be discounted (= reduced) at lower ranks. The typical discount is $\frac{1}{\log(rank)}$. The formula used to calculate DCG is:

$$DCG = r_1 + \frac{r_2}{\log_2 2} + \frac{r_3}{\log_2 3} + \cdots + \frac{r_n}{\log_2 n} =$$
$$= r_1 + \sum_{i=2}^{n} \frac{r_i}{\log_2 i}$$

This case uses base 2 for the logarithm, but any other base can be used as well. As for the other measures, it can be calculated only for the p top ranking documents instead of the whole set of retrieved ones.

An alternative formulation makes it so that high relevance judgements become much more important:

$$DCG = \sum_{i=1}^{n} \frac{2 * r_i - 1}{\log_2(1+i)}$$

This variant is used by some web search companies.

NDCG is evaluated over some number of k top search results; let Q be the set of queries and $R(j,d)$ the relevance score given to document d by query j. Then,

$$NDCG(Q,k) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} Z_{kj} \sum_{m=1}^{k} \frac{2^{R(j,m)} - 1}{\log_2(1+m)},$$

where Z_{kj} is a normalization factor calculated to make it so that a perfect ranking's NDCG at K for query j is exactly 1; the prefect ranking would be one that first returns all the docuents with the highest relevance level, then the next highest relevance level, and so on.

Bibliography