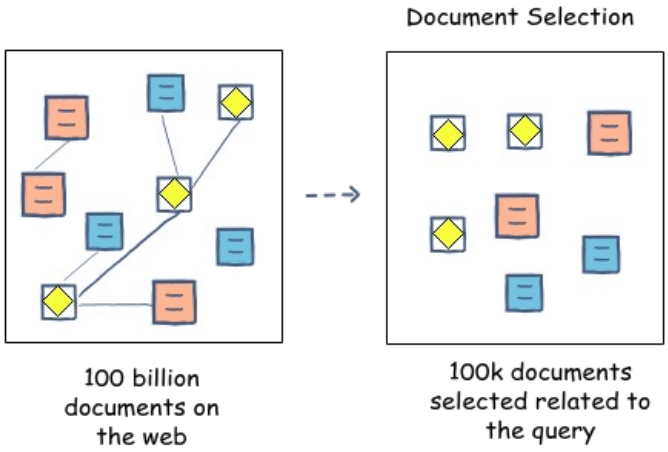
Document Selection

From the one-hundred billion documents on the internet, let's retrieve the top one-hundred thousand that are relevant to the searcher's query.

We'll cover the following

- Document selection process
  - Selection criteria
  - Relevance scoring scheme

Previously you saw the layered model approach. We will be adopting this approach to perform search ranking. Let's zoom in on the first step, i.e., *document selection*, as shown below:



The layered model approach

From the one-hundred billion documents on the internet, we want to retrieve the top one-hundred thousand that are relevant to the searcher's query by using *information retrieval* techniques.

Let's get some terminologies out of the way before we start.

> **Information retrieval** is the science of searching for information in a document. It focuses on comparing the query text with the document text and determining what is a good match.

Documents

Document types are as follows:

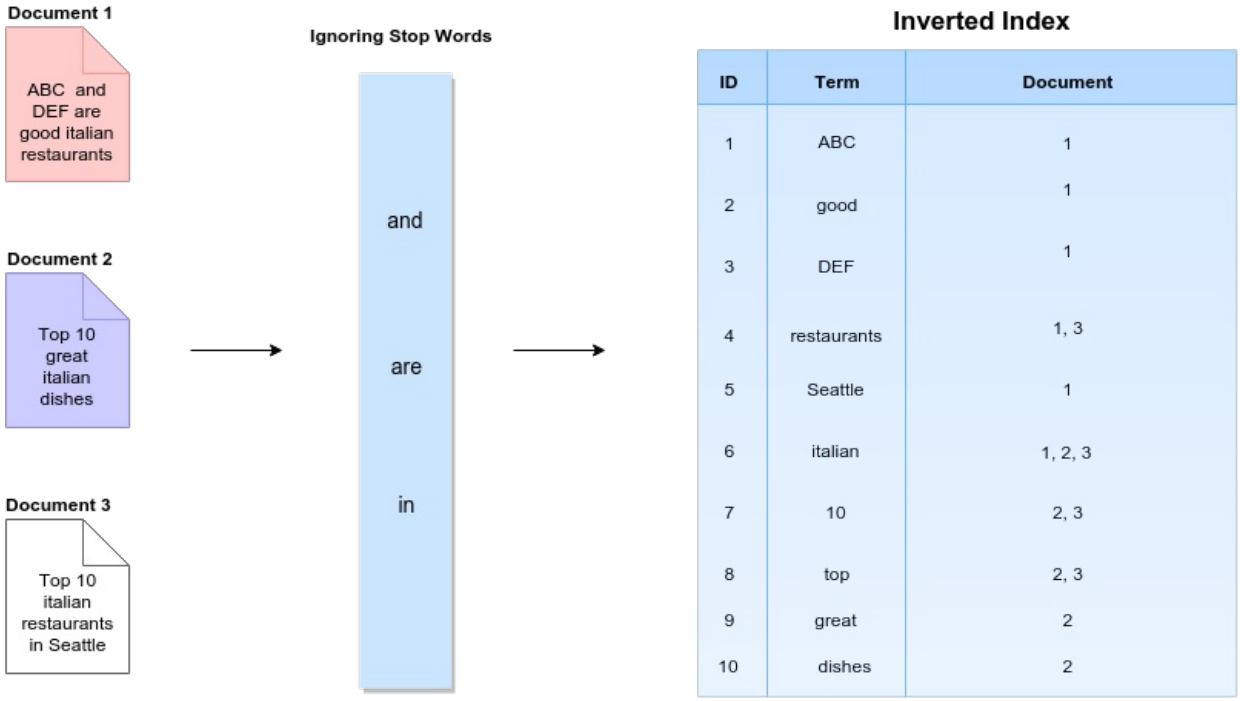
- Web-pages
- Emails
- Books
- News stories
- Scholarly papers

- Text messages
- Word™ documents
- Powerpoint™ presentations
- PDFs
- Patents, etc.

All of the above have a significant amount of textual content.

Inverted Index

> **Inverted index:** an index data structure that stores a mapping from content, such as words or numbers, to its location in a set of documents.

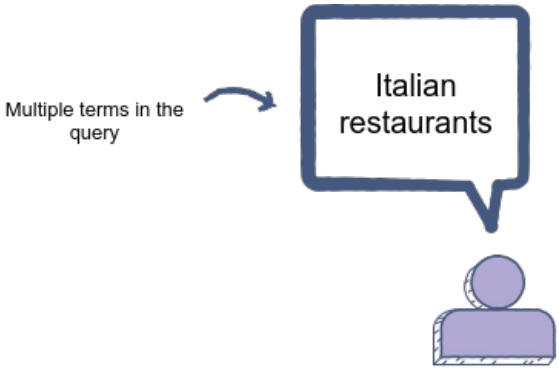


Inverted index: the term "restaurants" occurs in documents 1 and 3

Document selection process

The searcher's query does not match with only a single document. The selection criteria derived from the query may match a lot of documents with a *different degree of relevance*.

Let's begin by looking at the selection criteria and how it identifies matching documents. For instance, the searcher's query is:

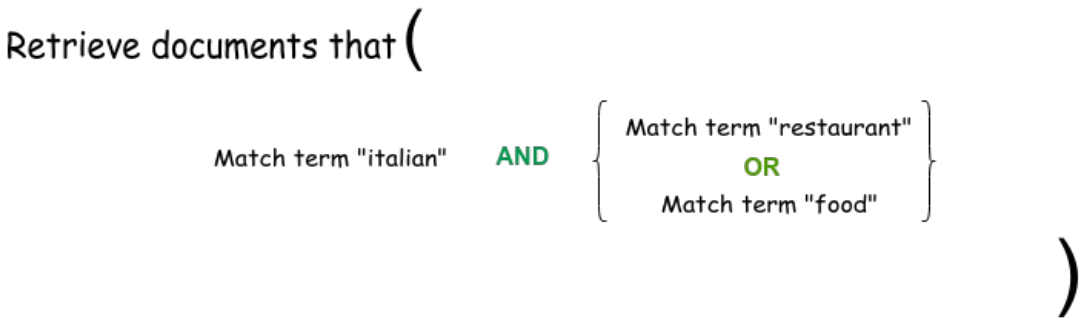


Searcher's query

The *query expansion* component tells us to look for *Italian food*, too.

Selection criteria

Our document selection criteria would then be as follows:



Selection criteria

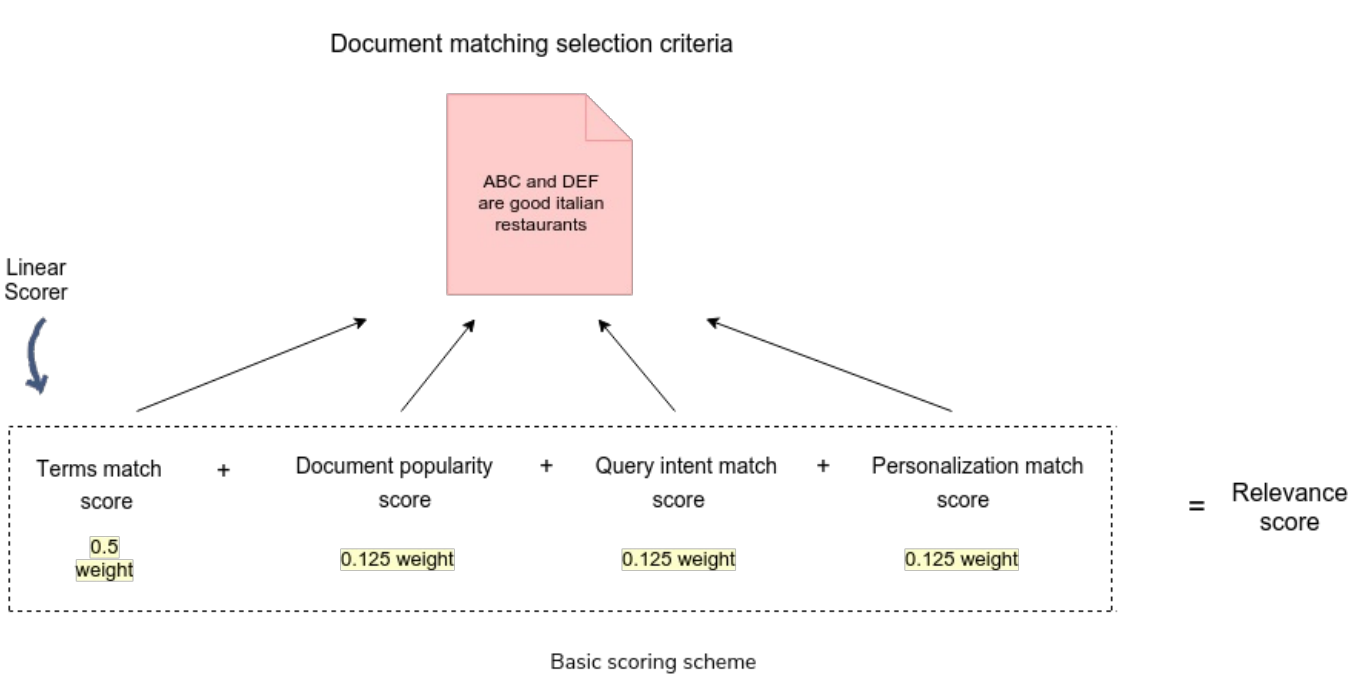
We will go into the *index* and retrieve all the documents based on the above selection criteria. While we would check whether each of the documents matches the selection criteria, we would also be assigning them a relevance score alongside. At the end of the retrieval process, we will have selected relevant documents sorted according to their relevance score. From these documents, we can then forward the top one-hundred thousand documents to the ranker.

Relevance scoring scheme

Let's see how the relevance score is calculated. One basic scoring scheme is to utilize a simple **weighted linear combination** of the factors involved. The weight of each factor depends on its importance in determining the relevance score. Some of these factors are:

1. Terms match
2. Document popularity
3. Query intent match
4. Personalization match

The diagram below shows how the linear scorer will assign a relevance score to a document.



> The weight of each factor in determining the score is selected manually, through the intuition, in the above scorer. Machine learning can also be used to decide these weights.

Let's look at each factor's contribution to the score.

Terms match

The term match score contributes with *0.5 weight* to the document's relevance score.

Our query contains multiple terms. We will use the inverse document frequency or IDF score (https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-39940-9_933) of each term to weigh the match. The *match for important terms in the query weighs higher*. For instance, the term match for "italian" may have more weight in the total contribution of term match to the document's relevance score.

Document popularity

The document's popularity score is stored in the index. Its value will be given 0.125 weight during the document's relevance calculation.

Query intent match

The *query intent* component describes the intent of the query. The document's match with the query's intent will contribute with 0.125 weight to the document's relevance calculation.

For our query, the component may reveal that there is a very strong *local intent*. Hence, a 0.125 weight will be given for the documents retrieved to be local.

Personalization match

This factor contributes with 0.125 weight to the document's relevance score. It scores how well a document meets the searcher's individual requirements based on a lot of aspects. For instance, the searcher's age, gender, interests, and location.

Remember, we can also use ML to assign these scores following a fairly similar process, which we will use in our ranking stage.

Back
Architectural Components

Next
Feature Engineering

Mark as Completed

Report an Issue
Ask a Question
(https://discuss.educative.io/tag/document-selection__search-ranking__grokking-the-machine-learning-interview)