# Information Retrieval Final Project

# Theoretical Review

## By Ron Keinan

## Table of Contents

# The Information Retrieval search engine

## The retrieving task

The task of retrieving data from a user-defined query has become very common in recent years, and it is the most popular activity on the internet (Homte et al., 2022).

Information retrieval (IR) in computing and information science is the process of obtaining information system resources that are relevant to an information need from a collection of those resources. Searches can be based on full-text or other content-based indexing. Information retrieval is the science of searching for information in a document, searching for documents themselves, and also searching for the metadata that describes data, and for databases of texts, images or sounds. The goal of a text retrieval system is to present the user with a set of items that will satisfy his or her information need.

Information retrieval can be accomplished simply and rapidly with the use of search engines. (Niwattanakul et al., 2013). This allows users to specify the search criteria as well as specific keywords to obtain the required results.

## How do search engines work?

Search engines can be described as an answer machines, users look for their queries using search engines, these engines in turn respond for the requested queries. Millions of requests per a second, days and nights are requested, and all this giant number of users are excepting to retrieve the most relevant results.

Search engine jobs characterized in two tasks (Homte et al., 2022):

- Retrieving the most relevant results to the query.
- Based on the popularity of the web sites, engines rank the retrieved results.

For search engines relevance has meaning furthermore than finding correct words within pages. When the era of search engines started, search engines were built on a very simple idea which is finding right words in pages (Homte et al., 2022), now with the giant increase of amount of data on the web this simple idea will not work anymore. The complexity of search engines increased over the years, since many factors effect on the relevancy. Engineers started to think in the popularity of sites, page, documents, the more popular sites, documents, or pages means the more valuable information within them (Homte et al., 2022). This way worked very well and efficiently in term of user's satisfaction degree. In order to determine the popularity, engineers developed many algorithms for this purpose instead of manual determination which is impossible to perform due to the giant number of sites, 12 documents etc. on the web. These algorithms relay on hundreds of factors to make their decision whether this site is popular or not.

Full-text search, i.e., finding documents with text that match the given keywords or sentences, has become the dominant form of information access, especially thanks to web search engines such as Google (Costa, 2021), which have a strong influence on how users search in other systems.

## Similarity Measures

Similarity measures play important roles in information retrieval (Bollegala et al., 2007). In order that a search engine should be able to choose the document closest to the query, a similarity function must be created that will calculate how similar the texts of the query and the document are.

Creating precise ranking models for a new type of search, such as for web archives, brings many challenges. First, it requires a comprehensive knowledge of users, as described in. Second, it is necessary to quantify user relevance with the proper ranking features, some of which need to be created or redesigned for the new context. Third, all features need to be combined into one ranking model optimized towards a goal (Costa, 2021).

There are many methods for ranking similarity, and I will present some of the methods (used in my project).

**BOW – bag of words**

The bag-of-words model is a simplifying representation used in natural language processing and information retrieval. In this model, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity.

After each document get its vector, the engine calculates dot product for each doc with the query vector - As more words from query appear in the doc – the result is higher, and the document is more similar/relevant.

BoW representation suffers from its intrinsic extreme sparsity, high dimensionality, and inability to capture high-level semantic meanings behind text data. Zhao & Mao (2017) attempted to deal with these Disadvantages and offers a new method, fuzzy Bag-of-Words (FBoW). FBoW adopts a fuzzy mapping based on semantic correlation among words quantified by cosine similarity measures between word embeddings. Since word semantic matching instead of exact word string matching is used, the FBoW could encode more semantics into the numerical representation. In addition, they proposed to use word clusters instead of individual words as basis terms and develop fuzzy Bag-of-Word Clusters (FBoWC) models. Document representations learned by the proposed FBoW and FBoWC are dense and able to encode high-level semantics. The results on seven real-word document classification datasets in comparison with six document

representation learning methods have shown that their methods FBoW and FBoWC achieve the highest classification accuracies.

**Jaccard coefficient**

The Jaccard index, also known as the Jaccard similarity coefficient, is a statistic used for gauging the similarity and diversity of sample sets. The Jaccard coefficient measures similarity between finite sample sets and is defined as the size of the intersection divided by the size of the union of the sample sets - Jaccard(A,B) = $|A \cap B| / |A \cup B|$.

A similarity measurement between keywords and index terms of 2 documents (or query and document) is essentially performed to facilitate searchers in accessing the required results promptly. Jaccard coefficient is one of the popular but yet simple similarity measurement method between words. (Niwattanakul et al., 2013)

**TF-IDF and BM25**

TF-IDF, Term Frequency Inverse Document Frequency (TF-IDF) is an algorithm that determine what words in a corpus of documents might be more favorable to use in a query. Ramos (Ramos, 2003) explains the mathematical framework of TF-IDF and explains that words with high TF-IDF numbers imply a strong relationship with the document they appear in, suggesting that if that word were to appear in a query, the document could be of interest to the user. Ramos also proved it's efficiency in experiment of document retrieval and noted a major shortcoming in this algorithm - that it does not link between synonyms and between identical words that differ in singular/plural.

BM25 is a popular transformation for TF to lower its influence. Has upper and lower limit so better than logarithm. It also normalizes the rank by average doc length to "punish" long documents that can fit many queries. (long doc also has more content so don't punish too much). BM25 is a popular and more efficient kind of TF-IDF algorithm (Whissell & Clarke, 2011).

**Doc2Vec**

Doc2vec is an NLP tool for representing documents as a vector and is a generalizing of the word2vec method. It doesn't only give the simple average of the words in the sentence. Like BOW, it is an implementation of VSM.

The common similarity function is cosine similarity, it measures the similarity between two vectors of an inner product space(measured by the cosine of the angle between two

vectors). It represents distance between vectors in the VSM that means similarity of the texts represented by the vectors.

Doc2vec used since its creation in 2014 for many tasks of identifying texts and retrieving them according to a desired topic. In 2016, Lee et al. used D2V to clusterize positive and negative sentiments with an accuracy of 76.4% (Sangheon et al., 2016). The same year, Lau and Baldwin showed that D2V provides a robust representation document, estimated with two tasks: document similarity to retrieve 12 different classes and sentences similarity scoring (Han Lau % Baldwin, 2016).
In 2019, Dynomant (Dynomant et al., 2019) trained a doc2ved model on a corpus of 16 million documents in order to vectorize them for similar document retrieval and proved its efficiency.

## Ranking the documents

After choosing a type of similarity function, and after inferring the vectors to the query and to all the documents according to the index, the Search engine calculate the ratio between the query and each of the documents according to the selected index, and print them in descending order of similarity (Homte et al., 2022). The higher a document appears, the more relevant it seems and satisfies the query, although it is a complex process (Costa, 2021).
Simply put - so far the information retrieval system has done its part. Of course, to produce an advanced and high-quality system, feedback on the quality of the documents must be produced at this stage, to allow the system to learn and improve its capabilities.

# References


Bollegala, D., Matsuo, Y., & Ishizuka, M. (2007). Measuring semantic similarity between words using web search engines. www, 7(2007), 757-766.

Costa, M. (2021). Full-text and URL search over web archives. In *The Past Web* (pp. 71-84). Springer, Cham.

Dynomant, E., Darmoni, S. J., Lejeune, É., Kerdelhué, G., Leroy, J. P., Lequertier, V., ... & Grosjean, J. (2019). Doc2Vec on the PubMed corpus: study of a new approach to generate related articles. arXiv preprint arXiv: 1911.11698.

Han Lau, J., Baldwin, T., 2016. An empirical evaluation of doc2vec with practical insights into document embedding generation. arXiv preprint arXiv: 1607.05368, 2016.

Homte, J. K., Batchakui, B., & Nkambou, R. (2022). Search Engines in Learning Contexts: A Literature Review. International Journal of Emerging Technologies in Learning (iJET), 17(2), 254-272.

Niwattanakul, S., Singthongchai, J., Naenudorn, E., & Wanapu, S. (2013, March). Using of Jaccard coefficient for keywords similarity. In Proceedings of the international multiconference of engineers and computer scientists (Vol. 1, No. 6, pp. 380-384).

Ramos, J. (2003, December). Using tf-idf to determine word relevance in document queries. In Proceedings of the first instructional conference on machine learning (Vol. 242, No. 1, pp. 29-48).

Sangheon Lee, Xiangdan Jin, and Wooju Kim, 2016. Sentiment classification for unlabeled dataset using doc2vec with jst. In Proceedings of the 18th Annual International Conference on Electronic Commerce: e-Commerce in Smart connected World, page 28. ACM, 2016.

Whissell, J. S., & Clarke, C. L. (2011). Improving document clustering using Okapi BM25 feature weighting. Information retrieval, 14(5), 466-487.

Zhao, R., & Mao, K. (2017). Fuzzy bag-of-words model for document representation. IEEE transactions on fuzzy systems, 26(2), 794-804.