A Brief Survey of Information Retrieval Models

Ali Abrahem Alnaied

The Higher Institute of Science and Technology Tajoura

Information Technology Dept.

Tripoli , Libya

a_alnaied@yahoo.com

Esam Basheer Albashri

The Higher Institute of Science and Technology Regdalin

Administrative and Financial Sciences Dept

Regdalin , Libya

Esamelbashri@gmail.com

Abstract

Retrieval models are one of the essential concepts in IR system. In this work, we will present some basic retrieval models that are used to find the top-k answers to a given user query. It is not a trivial task to select the most important/relevant results from huge amount of documents. One of the most important models used in IR systems is statistical model which includes the vector space model that was first proposed by Salton and McGill (1983). This model represents documents and queries as vectors in multidimensional space and computes the similarity between queries and documents. Then, documents are ranked according to this similarity score. Another popular model used to represent documents and queries is language modeling. This model was first proposed by Ponte and Croft (1998). It is a statistical language model such that each document is viewed as a language model. Statistical language model is a probability distribution over all possible words in a language. Finally we will also present learning to rank strategy which becomes very popular in recent years. This strategy learns a ranking function using implicit or explicit relevance data. In the following subsections, we give the details of each different approach.

Vector Space Model

Vector space model (VSM) represents documents and user queries by vectors in a multidimensional space. For instance: $d_j = (w_{1,j}, w_{2,j}, \ldots\ldots, w_{t,j})$, $q = (w_{1,q}, w_{2,q}, \ldots\ldots, w_{n,q})$. Each dimension corresponds to a term vocabulary that is used to build a list (index) of terms. Morphological analysis reduces different word forms to be indexed. In the VSM, a collection of documents can be represented in the VSM by term document matrix. Each cell in the matrix corresponds to the weight of a term in the document, if a term (word) appears

in the document than its value in the vector is none-zero, which means similarity does exist, and zero otherwise. "The performance of the VSM depends on the term weights scheme, that is, the functions that determine the components of the vectors" [10]. TF-IDF weighting scheme is the product of two statistics, term frequency (tf) and inverse document frequency (idf) to compute the weight of a term for a document, where tf is the number of times a term occurs in a document, Therefore, the tf weights for the terms can be defined as:

$$\mathrm{tf_{ij}} = \frac{\mathrm{f_{ij}}}{\mathrm{max_{ij}}}$$

Where $\mathrm{f_{ij}}$the number of is times term i appears in document j, and $\mathrm{max_{ij}}$ denotes the maximum frequency of a term in document j.
idf is measure of what portion of the document collection contains the term.

Therefore, the idf values for the terms can be defined as:

$$\mathrm{idf_i} = \log_2\left(\frac{\mathrm{N}}{\mathrm{df_i}}\right),$$

Where N is the total number of documents and i is term in collection, $\mathrm{df_i}$ the number of documents that contains the term i.

In general, the VSM is used to determine whether the document are relevant/matched or not relevant/not matched to the user's query. This task can be done by computing their cosine of the angle between two vectors (document and query vectors). If these vectors are related, the cosine of their angle will be one. If the vectors are orthogonal (unrelated), the cosine of their angle will be zero, which is defined as follows:

$$\cos(\theta) = \frac{d \,.\, q}{||d||\ ||q||}$$

Where d is a document vector, q is a query vector, θ is the angle between two vectors, $||d||$ is the norm of vector d, and $||q||$ is the norm of vector q. The norm of a vector is defined as following:

$$||q|| = \sqrt{\sum_{i=1}^{n} q_i^2}$$

The resulting scores can then be used to select the top-scoring documents for a query.

In the VSM for retrieval:

- Preprocessing which converts documents and query to a vector of all distinct terms.
- Construct document term matrix.
- Computing TF-IDF weights.
- Convert each document to a weighted vector and also construct the query vector and compute similarity scores between document and query vectors.
- Ranking the documents based on the similarity score, in descending order and return the top-k documents as the query results.

An example of the vector space retrieval model

d1: ارض قمر سماء "land moon sky"
d2: سماء نجوم فضاء "space stars sky"
d3: سحاب ارض شمس "sun land cloud"
q: ارض ارض شمس "sun land land"

There are three documents each of them contain terms, some appear only in one document and other appear in two documents, the similarity between each document and query can be computed as following steps:
Step1: We count the number of times of all distinct terms (word) appears in a document by calculate the tf scores for each word and their frequency on each of the document. We suppose the words in the vectors are ordered alphabetically.

	ارض	سحاب	سماء	شمس	فضاء	قمر	نجوم
D1	1	0	1	0	0	1	0
D2	0	0	1	0	1	0	1
D3	1	1	0	1	0	0	0

Table (2): Term Frequency

Step 2: We compute the term appears across the index by calculate idf for terms occurring in all the documents. The total number of documents is N=3.

Term	Idf
ارض	$\log_2 (3/2) = 0.58$
سحاب	$\log_2 (3/1) = 1.58$
سماء	$\log_2 (3/2) = 0.58$
شمس	$\log_2 (3/1) = 1.58$
فضاء	$\log_2 (3/1) = 1.58$
قمر	$\log_2 (3/1) = 1.58$
نجوم	$\log_2 (3/1) = 1.58$

Table (3): Inverse Document Frequency

Step 3: The weight of a term (w_{ij}) is the product of both tf_{ij} and idf_i which measure of the defined as step [1, 2]: $w_{ij} = tf_{ij} * idf_i$, we obtained the following matrix of documents as following:

	ارض	سحاب	سماء	شمس	فضاء	قمر	نجوم
D1	0.58	0	0.58	0	0	1.58	0
D2	0	0	0.58	0	1.58	0	1.58
D3	0.58	1.58	0	1.58	0	0	0

Table (4): tf.idf weights

Step 4: We calculate tf-idf vector for the query term. Therefore, we divide the frequency by the maximum frequency (2) than multiply with the idf values.

	ارض	سحاب	سماء	شمس	فضاء	قمر	نجوم
Q1	(2/2)*0.58 = 0.58	0	0	(1/2)*1.58 = 0.29	0	0	0

Table (5) query weight

Step 5: We calculate the length of document vectors and query vector.

Step 6: Compute the cosine similarity values between the query vector and each document vector by calculating the cosine of the angle between two vectors.
The similarity value between d1 and query is:
cos(d1,q) = (0.58*0.58 + 0*0 + 0.58*0 + 0*0.29 + 0*0 + 1.58*0 + 0*0) / (1.78*0.71) = 0.15
The similarity value between d2 and query is:
cos(d2,q) = (0*0.58 + 0*0 + 0.58*0 + 0*0.29 + 1.58*0 + 0*0 + 1.58*0) / (2.31*0.71) = 0
The similarity value between d2 and query is:
cos(d3,q) = (0.58*0.58 + 1.58*0 + 0*0 + 1.58*0.29 + 0*0 + 0*0 + 0*0) / (2.31*0.71) = .49

Ranking search results with respect to the query amongst three documents according to the similarity values. The ranking of documents based on decreasing cosine similarity will be: d3, d1, d2.

Language Model

Language model (LM) is a statistical model in the natural language process that is used to compute the probability of any context such as sentence or sequence of words. Typically, models that assign probabilities to sequences of words are called language model. "The language modeling approach to information retrieval has been successfully applied to many different applications because of its flexibility and theoretically solid background" [7]. Most of LMs approaches rank the documents by the probability of the query which is defined as: $P(Q|M_d)$. Each sentence or word sequences can gives different probabilities. Therefore, documents are ranked according to the highest probability to the query.
On the other hand, LMs are compute the probability of a sentence or sequence of word of length n, it assigns a probability P (w) = P (w1, w2,… . , w_n).

For instance: Given an Arabic sentence as follows:

P("عمر احمد علي") ("Ali Ahmed Omer")
P("جامعة عمر المختار") (" OMAR AL-MUKHTAR University")

In the LM shown above, the former phrase "عمر احمد علي" can has a higher probability than to "جامعة عمر المختار", because the words in the phrase "عمر احمد علي" occur more frequently in the query than the words in the others.

The LM steps can be defined as below:

- Estimate a document LM by estimating the probability of each word.

- Computing the query likelihood.
- Rank documents based on likelihood probability which is relevant of user's query.

N-grams

N-gram models are widely used in statistical natural language processing. They are basically a set of N items such as sentence or sequence of words.

$$P(w_1, \ldots, w_m) = \prod_{i=1}^{m} P(w_i \mid w_1, \ldots, w_{i-1})$$

N-grams are used for predict the previous words (N-1) in a sequence to predict the next word (N). An n-gram of size is one (N=1), this is called "unigrams" and this is essentially the individual words in a sentence, when (N=2) this is called "bigrams", when N=3 this is called "trigrams", and when N>3 this is sometimes referred to as four grams or five grams and so on. These known as LMs type which can be defined as following formula:

Unigram: $P(w_{1,n}) = P(w_1)P(w_2) \ldots . P(w_1)$ (1)
Bigram: $P(w_{1,n}) = P(w_1)P(w_2|w_1) \ldots . P(w_n|w_{n-1})$ (2)
Trigram: $P(w_{1,n}) = P(w_1)P(w_2|w_1)P(w_3|w_{1,2}) \ldots . P(w_n|w_{n-2,n-1})$ (3)

For instance, the sentence "Welcome to OMAR AL-MUKHTAR University Al-Bayda"

If N=2 (bigrams) Thus, we have five n-grams in this case as following:

Welcome to
to OMAR
OMAR AL-MUKHTAR
AL-MUKHTAR University
University Al-Bayda

If N=3 (trigrams) Thus, we have four n-grams in this case as following:

Welcome to OMAR
to OMAR AL-MUKHTAR
OMAR AL-MUKHTAR University
AL-MUKHTAR University Al-Bayda

1 Unigram LM

Unigram LM (which also known as the bag of words model) is a probability distribution over individual words included in the text, it means it does not depend on any of its previous words. Unigram computes as: Probability of word i = Frequency of word (i) / total number of words.

Example 1:
The probability of the sentence "Sea River Lake Ocean"
By reference to the formula No.1 we compute probability of "Sea River Lake Ocean" as below:
P (Sea, River, Lake, Ocean) = P(Sea) P(River|Sea) P(Lake|Sea, River) P(Ocean|Sea, River, Lake).
Example 2:
Document "university university university of Al-Bayda Al-Bayda Al-Bayda Al-Bayda"
Above document contain eight words, the probability of each word independent as below:

P(university) = 3/8
P(of) = 1/8
P(Al-Bayda) = 4/8

P(university of Al-Bayda) = P(university) x P(of) x P(Al-Bayda) = 3/8 x 1/8 x 4/8 = 0.023

P(Al-Bayda university) = P(Al-Bayda) x P(university) = 4/8 x 3/8 = 0.019

On the other hand, if a term in a query does not appear in a document then the probability value become zero. In this case, smoothing methods should be applied to avoid zero probability problem.

2 Bigram LM

Bigram, also called Markov assumption, assumes that we can predict the probability of the next word by only looking at the previous word. In other words, instead of computing the probability.

P(Welcome| to OMAR AL-MUKHTAR university Al-Bayda)

We approximate it with the probability

P(Welcome| Al-Bayda)

In general, a bigram is an n-gram for N=2, this model use to predict the conditional probability of the next word. The simplest way to estimate bigram probability by using Maximum Likelihood Estimation (MLE), based on taking counts from the corpus and normalizing them. The general equation for estimating probability for a MLE Bigram can be defined as below:

$$P(W_i \mid W_{i-1}) = \frac{\mathrm{Count}(W_{i-1}, W_i)}{\mathrm{Count}(W_{i-1})}$$

Where $\mathrm{Count}(W_{i-1}, W_i)$ is the count of the occurrence of the word W_{i-1} followed by word W_i, and $\mathrm{Count}(W_{i-1}, W_i)$ is the count of the occurrence of the words in corpora.

3 Trigram LM

Trigram is an n-gram for N=3, as in bigram model, Trigram model computes the probability of the next word depends only on the previous two words, $such\ as\ \mathrm{P}(\mathrm{W}_3 \mid \mathrm{W}_1, \mathrm{W}_2)$, which is defined as the formula No.3.

For example:
Document “Sky is blue”.
To compute the probability of the word blue following the words Sky is. The MLE of this trigram probability is:

$$\mathrm{P}(blue \mid \mathrm{Sky\ is}) = \frac{\mathrm{Count(Sky\ is\ blue\ )}}{\mathrm{Count(Sky\ is)}}$$

Therefore, we need to count of the occurrence of the trigram Sky is blue in the document as well as the count of the bigram history Sky is. When building smoothed trigram LM's, we also need to compute bigram and unigram probabilities.

Query likelihood LM

The query likelihood model is a basic method for using LM in an IR system. It constructs query likelihood from each document in the collection. Documents are then ranked based on the highest probability to the query in the document's LM $\mathrm{P}(\mathrm{q} \mid \mathrm{M_d})$. Typically, the unigram LM is used in this case. The query likelihood model can be defined as following:

$$\mathrm{P}(q \mid \mathrm{d}) = \frac{\mathrm{P(q|d\ )P(d)}}{\mathrm{P(q)}}$$

Where P(q), probability of the query, is the same for all documents and P(d) is probability of the document.

Smoothing

There are several ways of avoiding zero probabilities and making the estimates for n-grams, these are known as smoothing methods. Smoothing is a technique to give some probability massing (unseen) words to n-grams that have not occurred in the corpus. Therefore, probability mass has to be taken from n-grams that occur in the corpus.

Learning to Rank

With the fast development of the Web and the difficulties in finding needs information. The search engine has become the most important tools for many people. The ranking problem is a central component (Components such as TF, TF-IDF, Lengths, IDF, and Document's PageRank are called features) in every search engine, its goal to produce a ranked list of documents

according to the relevance between processed queries and indexed documents. I recent decade, different machine learning technologies have been used to train the ranking model, and a new research area named (Learning to rank). It has become one of the most technique used in IR systems to solve the problem of ranking methods. Learning to rank combines various features based on document weighting models included documents represented by feature vectors and discriminative training which an automatic learning process. In addition to that, learning to rank is the ability to use multiple features in a uniform way during ranking and learn an appropriate method to combine those features.

Conclusion

Retrieval models play a fundamental role in influence extends to how textual data is represented, matched, and ranked, making them critical components in the design of efficient retrieval systems. The ongoing development and evaluation of these models are essential for advancing IR capabilities, especially in the face of growing and increasingly complex data environments. This study explored the impact of three prominent retrieval models—Vector Space Model, Language Model, and N-gram Model—on IR performance. Our findings highlight the strengths and limitations of each approach, emphasizing the importance of selecting the appropriate model based on the nature of the data and the retrieval context. Future research should continue to investigate hybrid approaches and model optimizations to further improve accuracy, relevance, and efficiency in IR systems

References

1. Total number of Websites (March 14, 2016) From ınternet lıve stats. http://www.internetlivestats.com/total-number-of-websites/

2. Aljlayl, M. and Frieder, O. “On Arabic Search: Improving the Retrieval Effectiveness via a Light Stemming Approach”, ACM Eleventh Conference on Information and Knowledge Management, Mclean, VA, November, 2002.

3. Larkey, L.S.; Ballesteros, L.; and Connell, M.E. (2002). Improving stemming for Arabic information retrieval: Light stemming and co-occurrence analysis. SIGIR '02Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval.

4. G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. Information Processing and Management, 24(5):513-523, 1988.

5. J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 275–281, 1998.

6. al-Jlayl, M., & Frieder, O. (2002). On Arabic search: Improving the retrieval effectiveness via light stemming approach. Proceedings of the 11th ACM International Conference on Information and Knowledge Management, Illinois Institute of Technology. New York: ACM Press, 340-347.

7. Hmeidi, I., Kanaan, G., & Evens, M. (1997). Design and implementation of automatic indexing for information retrieval with Arabic documents. Journal of the American Society for Information Science, 48(10), 867 881.

8. Abu Salem, H. (1992). A microcomputer based Arabic bibliographic information retrieval system with relational thesauri. Unpublished doctoral dissertation, Computer Science department, Illinois Institute of Technology, Chicago.

9. al-Kharashi, I. (1991). Micro-Airs: Microcomputer based Arabic information retrieval system, comparing words, stem, and roots as index terms. Unpublished doctoral dissertation, Computer Science department, Illinois Institute of Technology, Chicago.
10. Mustafa, Suleiman H. (2007). Word stemming for Arabic: The case for simple light stemming: The 8th International Arab Conference on Information Technology (ACIT 2007), pp. 396-410. Lattakia, Syria, 26-28, Nov. 2007.

11. Rowley, Jennifer (1998). The electronic library. London: Library Association Publishing.

12. Webology, (April 14, 2016), http://www.webology.org/2006/v3n1/a22.html.

13. Larkey, S. L., Ballesteros, L., and Connell, E. M. (2005), “Light stemming for Arabic information retrieval”. Arabic Computational Morphology: Knowledge-based and Empirical Methods.

14. Mohamad Ababneh, Riyad Al-Shalabi, Ghassan Kanaan, and Alaa Al-Nobani "Building an Effective Rule-Based Light Stemmer for Arabic Language to Improve Search ffectiveness" The International Arab Journal of Information Technology, Vol. 9, No. 4, July 2012

15. Riyad Al-Shalabi, Ghassan Kanaan, Sameh Ghwanmeh, "Stemmer Algorithm for Arabic Words Based on Excessive Letter Locations" 978-1-4244-1841-1/08/$25.00 ©2008 IEEE

16. Ahmed Khalid, Zakir Hussain, Mirza Anwarullah Baig, "Arabic Stemmer for Search Engines Information Retrieval" IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 1, 2016.

17. Asma Al-Omari, Belal Abuata, ARABIC LIGHT STEMMER (ARS), Yarmouk University, 21163 Irbid – Jordan, 2014. Journal of Engineering Science and Technology December 2014, Vol. 9(6)

18. L. S. Larkey, L. Ballesteros, M. E. Connell, “Improving stemming for Arabic Information Retrieval: light stemming and co- occurrence analysis,” Proc. of the 25th SIGIR, Tampere, Finland, 2002, pp. 275-282.

19. Mourad Gridach , Noureddine Chenfour, Developing a New Approach for Arabic Morphological Analysis and Generation. Mathematics and Computer Science Department, Sidi Mohamed Ben Abdellah University – Faculty of Sciences, Fez, Morocco