Information retrieval approaches: A comparative study

Case Study

Assmaa Moutaoukkil
IRF-SIC Laboratory, Department of Computer Science, Faculty of science,
Ibn Zohr University, Agadir, Morocco
Asmae.mt1@gmail.com

Ali Idarrou
IRF-SIC Laboratory, Department of Computer Science, Faculty of Science,
Ibn Zohr University, Agadir, Morocco
aliidarrou@gmail.com

Imane Belahyane
IRF-SIC Laboratory, Department of Computer Science, Faculty of Science,
Ibn Zohr University, Agadir, Morocco
imane.belahyane@edu.ac.uiz.ma

Abstract *– The area of information retrieval (IR) has taken on increasing importance in recent years. This field is now of interest to large communities in several application domains (security, medicine, aeronautics, etc.). IR studies find relevant information from the semi-unstructured type of data. As the information resources generated after the search can be extensive in quantity and different in quality, it is essential to rank these results according to the degree of relevance. This paper focuses on text information retrieval (TIR) and emphasizes the importance of each IR approach. This study presents insightful aspects of TIR and provides a comparative study between some proposed approaches and models. Each model offers IR advantages and suffers from several limitations.*

Keywords*: information retrieval, NLP, graph, TF-IDF, document classification, KNN, fuzzy K-means, deep learning, BERT, transformers,*

1. INTRODUCTION

The increasing growth of technologies, storage capacity and computational power, and modification of telecommunication technologies provided to deal with different types of information maintained by different types of media. The amount of online data has grown at least as quickly as the speed of computers, and we would now like to be able to search collections that total in the order of billions to trillions of words. Consequently, several procedures and methods are needed to find the relevant information at the right time.

Information has always been the raw material of all processing for the benefit of organizations and individuals. It is created, processed, shared, stored, and transmitted by these units loudly. The continuity and development of any legal or physical person or unit are linked to its ability to access relevant information at the right time. While the digital revolution has undoubtedly reduced the importance of time and distance, in some ways, it has grown it in others: IR and managing large corpora and collections have become significant challenges. Many applications that deal with information would be inadequate without the support of IR technology. IR had occupied information scientists before the term "information science" was coined [1]. Similarly, information science, such as text mining, human-machine interaction, NLP, machine learning, Big Data, etc., is in the service of IR. Moreover, if the information is an integral part of today's world, IR is a decisive activity for the people who inhabit it. The tasks related to this domain seek to automatically respond to the information needs of a user who hopes to solve a problem or achieve a goal for which the current state of his knowledge is inadequate [2]. If the idea is simple, the methods are much more delicate, especially when dealing with complex data, for example, by their modality, as is the case of the textual contents which constitute the object of study of this work. One way to clarify the problems treated in this work is to ask these questions:

- How to capture and model the need initially expressed by the user to fully satisfy it?
- How can content be optimally represented to facilitate the IR process?
- How to access the relevant documents requested by the user through a query in an IR system?

- How to compare and class documents in a dynamic, rich, unstructured, and voluminous collection?

There is no single convincing answer to these questions. There are many approaches, called here models, and each is useful for developing some IR tools. In sections 4,5,6, and 7, we will explain in a pedagogical style, respectively, the models based on bag-of-words representation, those based on graphs, those based on classification techniques, and those based on deep learning techniques. We will devote section 8 to discussing and comparing these models. We explain in section 3 the methodology followed for the elaboration of this work. But first, we will describe what exactly these models denote (section 2).

2. INFORMATION RETRIEVAL(IR)

IR refers to the process, methods, and procedures for locating and extracting data and information stored in a semi-structured or unstructured database.

Gerard Salton defined IR as follows: "Information retrieval is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information." [3]

IR is the science of searching for information in a document or documents themselves and searching for the metadata that describes the data and databases of text, images, or sounds.IR is the science of obtaining information system resources relevant to an information need from a collection of such resources. In this paper, we deal with textual data only.

2.1. IR PROCESS

Information Retrieval Systems (IRS) are concerned with search strategies in which the retrieved documents may be more or less relevant to the query. There are four basic processes that an IRS must support: *indexing*, *interrogation*, *similarity evaluation*, and *ranking*. These processes are visualized in Fig. 1, where the square boxes represent objects, and the rounded boxes represent processes.

The process of IR begins with the representation of information. The elementary object of information storage is the document. The representation of documents is generally referred to as the indexing process. This process takes place offline, i.e., the end-user of the IRS is not directly involved. The indexing process results in a representation of the document in the first step (1) the represented documents are then stored (2) the query, in turn, must be represented in the same way as the documents (5)(6) The IRS then proceeds to a comparison process through a similarity measure which changes depending on the IR model used (7)(8). The ranking process aims at sorting the documents deemed relevant by the system (9). The user can then judge the relevance of the returned documents based on either his initial information need or a new information need. An interrogation process assists the user in making his vague and ambiguous need explicit in the form of a query (3).

In order to optimize the results of an IRS, a query expansion process is always recommended, either by explicit or implicit feedback.

3. METHODOLOGY

This work was conducted following the guidelines suggested by [4][2][5]. The process followed is illustrated in Fig. 2. It should be noted that it is sometimes difficult to compare the results published in different papers due to incomplete or insufficient information about the algorithm used by the authors, e.g., the pre-processing methods adopted may lead to significant differences. Therefore, whenever possible, we collect results from papers that contain comparisons between some of these models performed on a single site for reliability reasons. We compare the empirical evaluation results of the most prominent IR models previously discussed on several popular reference datasets. Nevertheless, some previous studies are not based on popular corpora, and in this case, our analysis is based on other additional criteria, namely:

- The nature, number, and category of the documents in the collection.
- We have also specified criteria for selecting papers to facilitate comparing these models (comparison of the proposed contribution with previous models, ...)

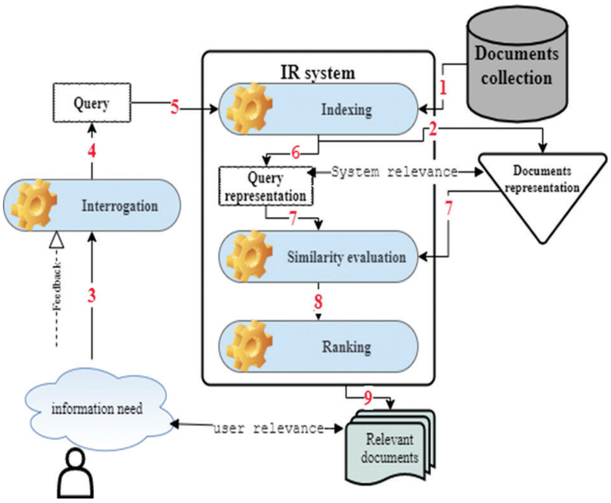

Fig. 1. IR process

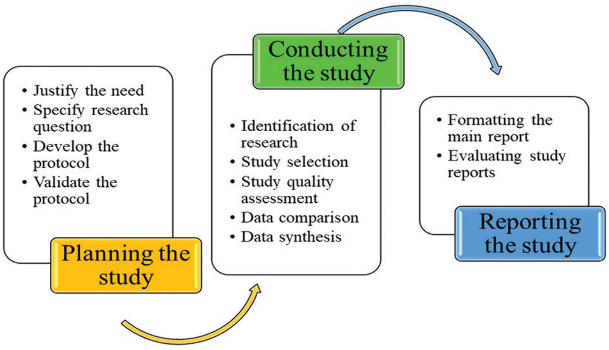

Fig. 2. Study process

4. MODELS BASED ON THE BAG-OF-WORDS REPRESENTATION

What is important to users is the content of a document. Given that the content is semi-structured or unstructured, the challenge for an IRS is to produce a set of words or terms that are sufficiently descriptive to represent the content of a document, in addition, to measuring the importance of each word in a particular document and the collection in general.

4.1 BAG OF WORDS REPRESENTATION

The initial idea is to use keywords to identify the content of a document[6] as well as the content of the query. The process starts by listing all the significant words in each document to identify the keywords in the collection. The goal is to find the words that best describe the content of a document.

Bag-of-words representation provides a way to organize the textual content of a collection of documents into a matrix (Table 1) in the hope of mapping key terms (words) to documents to measure the importance of each term in each document in the collection using weighting techniques.

Table 1. bag of words representation

	d1	d2	....	dn-1	dn
t1	$W_{1,1}$	$W_{1,2}$	.....	$W_{1,n-1}$	$W_{1,n}$
t2	$W_{2,1}$	$W_{2,2}$	...	$W_{2,n-1}$	$W_{2,n}$
⋮ ⋮	⋮ ⋮ ⋮	⋮ ⋮ ⋮	⋮	⋮ ⋮ ⋮	· ⋮ ⋮ ⋮
tn-1	$W_{n-1,1}$	$W_{n-1,2}$	....	$W_{n-1,n-1}$	$W_{n-1,n}$
tn	$W_{n,1}$	$W_{n,2}$	....	$W_{n,n-1}$	$W_{n,n}$

In order to assign a weight to each term, many weighting solutions have been proposed in the literature [7].The most impressive is the TF-IDF. Furthermore, since a demonstration is better than a long explanation, we present the following example:

D {d1,d2,.......,dn} a collection of documents.

T {t1,t2,.......,tn } of terms belonging to this collection without redundancy.

$\mathbf{W_{t,d}}$ The statistical measure that reflects the importance of a term t in document d.

4.2 WHY THE TF-IDF METHOD?

TF-IDF is a weighting method that measures the importance of a word in a document relative to a collection of documents. It is a statistical measure that is based on the calculation of the weight of each term in a document relative to a collection of documents according to the following general formula:

$$W(t,d) = TF \times IDF \tag{1}$$

TF (Term frequency) is the frequency of appearance of a term t in document d. The number of times that t appears in d. many variants have been proposed in the literature. The simplest is the Boolean model; 0 if the term is present in document 1 otherwise, this approach is very limited. Others propose a raw frequency or a logarithmic normalization to dampen the differences or a normalization that considers the length of the document.

The IDF has a discriminating power. IDF (inverse document frequency) measures how common or rare a term is in a collection of document D. It allows assigning a high weight to less frequent terms and a lower weight to more frequent terms. Fig. 3 and 4 show the results of an experiment that proved the usefulness of using both TF and IDF [8]. The collection studied was 1400 documents encoded in SGML text format, keeping the formatting tags to account for noisy data and to test the robustness of TF-IDF. The authors of [8] apply case sensitivity to simulate more noise. The proposed SRI returns the first 100 documents as follows:

It is quite clear that TF-IDF is more powerful than TF alone. It provides high values for rare words and low values for common words. TF returns imprecise results; relevant documents are scattered sporadically. This can be explained by IDF eliminating the more frequent stop-words.

The results in [9] show that the TF-IDF system managed to have an accuracy that exceeds that achieved by the TF-ATO system in the first case without a discriminative approach or stop-word removal. This is explained by the fact that the TF-IDF system can remove insignificant terms (IDF).

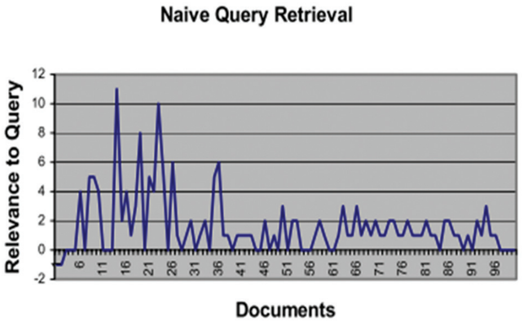

Fig. 3. Result using TF alone to calculate the weights [8]

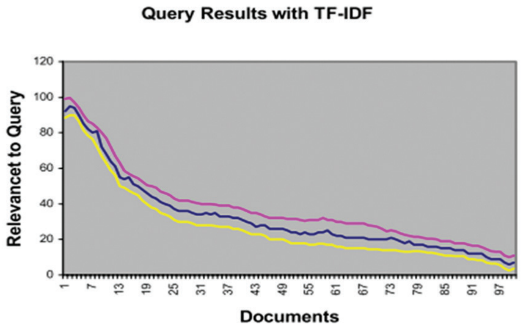

Fig. 4. Result using TF-IDF to calculate weights [8]

4.3 PREPROCESSING DATA TO IMPROVE PERFORMANCE

In written language, some terms carry more semantics than others. Several authors claim that there is still interest in improving the simple TF-IDF. Looking at the data, the simplest way to improve TF-IDF would be to ignore case sensitivity [8] and to use techniques to clean and normalize the data, such as Stemming: which is used to group the different forms of a particular word as well as to assimilate the words with their lexical derivatives and synonyms.

To limit the amount of useful information and the term space, several authors propose a preprocessing process to eliminate all non-significant words or reduce any other noise (upper case, lower case, suffix, prefix). As an example, the authors of [10] propose to remove stop-words such as "and", "a", and "an" because they are non-significant words, not having a discriminating character. They do not make it possible to distinguish one text from another.

Indexing is a crucial step in the IR process. In a keyword-based approach, this technique allows representing a document and/or a query by a set of keywords, also called descriptors. These descriptors constitute an exploitation facility of the documents by the IRS. Given the large amount of data to be processed by this system, preprocessing techniques can reduce the limits of this approach by cleaning the data, such as eliminating stop words, the stemming of Porter, Tokenization, etc. Nevertheless, preprocessing is not sufficient to increase the performance of an IRS.

5. GRAPH-BASED MODELS

A standard approach to IR is to model the text as a bag of words. Alternatively, the text can be modeled as a graph, whose vertices represent words and edges represent relationships between words, defined based on any meaningful criterion.

With such a graph, graph-theoretic calculations can be applied to measure various properties of the text.

The proposed idea is to transform IR into a graph problem. The graph data structure allows organizing and especially linking a set of objects (documents, terms, queries...) simply and practically.

A variety of models and techniques have been proposed for this purpose:

In [11] the authors present the graph-based Text-Rank model for extracting keywords and phrases from raw data.

The algorithm considers the weight of edges when computing the score associated with a graph vertex.

In [12] the authors introduce the use of a random walk algorithm to weight the terms in the TF-IDF weighting scheme by adapting the Text Rank algorithm. In this model, the edge weights are not considered.

In [13] the authors exploit the relationship between the local information of a vertex (term position) and the global information (information gain) and term dependence to produce term weights.

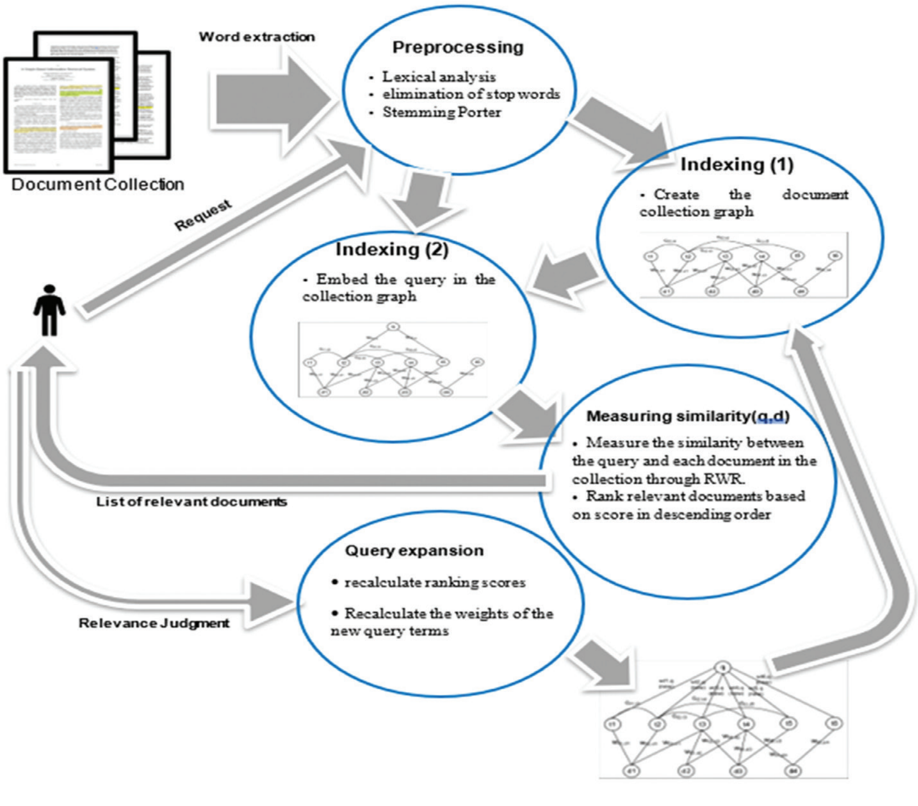
Word extraction
Preprocessing
• Lexical analysis
• elimination of stop words
• Stemming Porter
Document Collection
Request
Indexing (1)
• Create the document collection graph
Indexing (2)
• Embed the query in the collection graph
Measuring similarity(q,d)
• Measure the similarity between the query and each document in the collection through RWR.
• Rank relevant documents based on score in descending order
List of relevant documents
Query expansion
• recalculate ranking scores
• Recalculate the weights of the new query terms
Relevance Judgment

Fig. 5. IR graph-based model

For example, we will look at the model proposed in [14]. This model adopts a graph structure that captures the occurrences of terms in documents and the correlations between terms. The similarity between a document and the query is computed by a specific type of random walk graph algorithm called Random Walk with Restart (RWR). The model can be extended to incorporate relevant feedback naturally. Its effectiveness is evaluated using TREC collections; its performance is measured and compared to systems participating in the TREC program. The diagram in Fig.5 summarizes the proposed process.

The model builds a collection graph that represents the content of documents and the relationships between terms created before a query exists. When a user provides a query to the system, a query node is constructed and connected to the query terms in the graph. A graph random walk algorithm is then applied to calculate the relevance of each document to the query. The present model was tested on the TREC 2003 High Accuracy Retrieval from Documents (HARD) corpus (National Institute of Standards and Technology, 2009), which provides 20 training topics with 100 documents judged for each topic and uses a set of 48 topics for evaluation. The soft evaluation judgment is adopted in this experiment since the author does not use any additional information provided.

Four GIR configurations are studied:

The GIR-NoRelNoSig (Baseline) configuration uses binary link weights and does not include a relationship between terms.

The GIR-RelNoSig (corrBinary) configuration includes relationships between terms with binary link weights.

The GIR-RelSig (Full) configuration includes a term relationship with real-valued connection weights between 0.0 and 1.0.

The GIR-RF configuration is GIR-RelSig (RF) with the automatic relevance return process. The initial run's first ten documents are automatically fed into the model.

Comparing the four configurations of the GIR model, the performance was better from the weakest to the strongest GIR-NoRelNoSig, GIR-RelNoSig, and GIR-RelSig, respectively.

6. CLASSIFICATION TECHNIQUES BASED MODELS

The idea here is to transform the IR problem into a classification problem. As shown below in Fig. 6, the query is considered a document; we have a set of determined classes. Once a new document arrives, we must put this document in the suitable class or classes.

Dynamic document classification reduces effort and time as it processes the new document and assigns it directly to the appropriate classes without re-running the entire algorithm. [15]

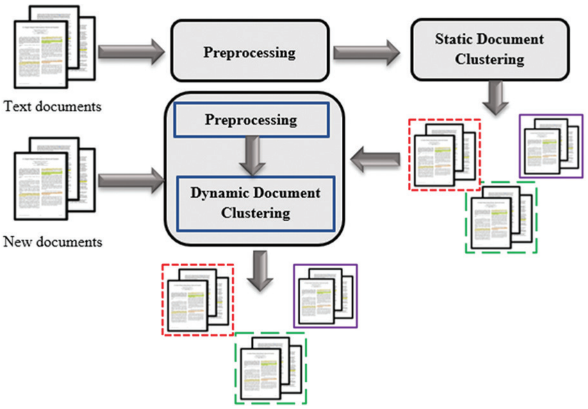

Fig. 6. A general process of dynamic document classification

Classification as a learning method automatically assigns one or more classes to a free text document. Document classification can be performed in supervised or unsupervised learning mode: "Supervised classification aims to associate each of the n observations {x1, ..., xn} with one of the k classes known a priori while unsupervised classification aims to group these data into k homogeneous groups" [16]

6.1 MODEL-BASED ON SUPERVISED CLASSIFICATION

Supervised classification is based on input and output examples to classify new documents. Predefined classes are assigned to text documents based on their content. The typical text classification process consists of *preprocessing*, *indexing*, *dimension reduction*, and *classification* [17]. The goal of classification is to train a classifier based on known examples, and then unknown examples are assigned automatically. Statistical classification techniques such as Naïve Bayesian classifier, nearest neighbor classifier, decision tree, and support vector machines can be used to classify the text. In [18] authors proposed a combination of TF-IDF and the KNN algorithm for classification. The optimized method performed six times better than the conventional method. Nevertheless, the quality of classification decreases slightly with the increase in the number of documents. The studied collection is 500 documents of different lengths and different categories. The following table (Table 2) summarizes the implementation of this algorithm and the result of this classification:

Table 2 . KNN and TF-IDF

Representation	**Bag of words**			
Preprocessing	⇨ adjust documents to classic text format ⇨ automatically remove control characters			
TF-IDF Formula	$a_{ij} = tf_{ij}idf_i = \frac{f_{ij}}{\sqrt{\sum_{s=1}^{N}\left(tfidf(a_{sj})\right)^2}} \times log_2\left(\frac{N}{df_i}\right)$			
Measure of similarity	**Euclidean distance**			
Documents categories	Sport	Politics	Finance	Daily news
Classification quality	0,92%	0,90%	0,78%	0,65%

This algorithm presents accurate results for the category "*sport*" the main reason for the excellent classification results in this category is that the documents were not textually demanding. The documents did not contain many different words, which reduced the impact of unusable words and characters. On the other hand, the worst classification was for the "*Daily News*" category. The content analysis of the documents in this category showed that these documents contained a lot of *unusable words*, those words that are often repeated and do not have a significant weight but harm the classification.

6.2 MODEL-BASED ON UNSUPERVISED CLASSIFICATION

Unsupervised classification or clustering is used to group similar documents. This method is based on the concept of dividing a similar text into the same cluster. Each cluster contains many similar documents [19] . The objects are grouped or clustered based on maximizing intra-class similarity and minimizing inter-class similarity.[20]

The authors of [15] proposed a comparative experiment between the two clustering techniques, HAC (hierarchical agglomerative clustering) and fuzzy K-means, for the same collection of 45 documents of different categories. The following table (Table 3) summarizes the algorithm followed along and the results obtained.

Table 3. HAC Vs fuzzy K-means

Representation		Bag of words			
Preprocessing		⇨ Eliminate StopWords ⇨ Stemming			
TF-IDF Formula		Nothing to report			
Measure of similarity		Cosine distance			
Documents categories		News 20	Reuters	Research papers	E-mail
HAC	Entropy	0.256213	0.112368	0.678951	0.225641
	F Measure	0.8612 31	0.8890 23	0.5234 10	0.7456 12
Fuzzy K-means	Entropy	0.389763	0.421189	0.245612	0.214561
	F Measure	0.785621	0.614523	0.884532	0.754312
Number of Clusters		4	6	4	4

It appears that HAC *New 20* and *Reuters data* perform better. On the other hand, Fuzzy K-means is more accurate for the Research Paper data. Either algorithm can be applied to the E-mail dataset, as the entropy and accuracy values do not show a significant difference. However, the graph (Fig. 7) indicates that e-mails can be well classified with Fuzzy k-means, which produces non-overlapping clusters.

Thus, the impact of document categories on the performance of either supervised or unsupervised classification is infallibly proven.

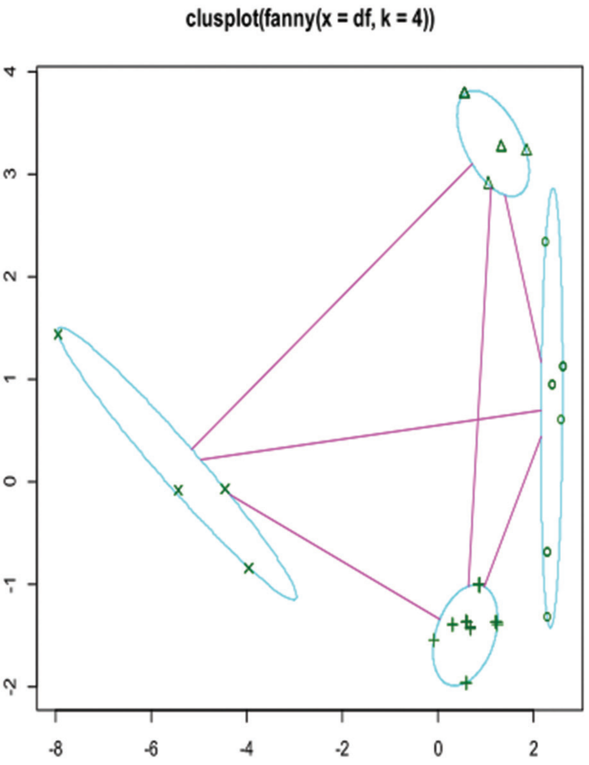

Fig. 7. Fuzzy K-means plots for email dataset[15]

6. 3 CHOOSING THE RIGHT SIMILARITY MEASURE

An experiment in [21] aims to find the best distance measure of hierarchical clustering techniques (HAC), K-means, and K-medoids. Many distance measures were tested: 'Euclidean distance', 'Cosine distance', 'Distance between neighborhoods', 'Hamming distance', 'Jaccard distance', 'Correlation distance', 'Chebyshev distance' and 'Sqeuclidean distance'.

The collection comprises five categories: Business, Education, Elections, Entertainment, and Games. Each category has 50 articles selected for implementation, which are taken from the websites of different news channels. In hierarchical clustering, Euclidean distance measure by means was the best. In K-means clustering, correlation distance measure, and K-medoids clustering, city block distance measure shows better results than other distance measures.

The HAC evaluation results show that the *classification quality using Euclidean distance exceeds that of Cosine distance by 0.2597*. Hence, the usefulness of questioning the results of using Euclidean similarity measure in the HAC technique of the algorithm proposed by [15].

7. DEEP LEARNING BASED MODELS

For text classification, after classification learning came deep learning. In the context of IR, deep learning approaches were attractive for two reasons: First, continuous vector representations freed text retrieval from the limitations of exact term matching. Second, neural networks promised to avoid the need for laboriously hand-crafted features (which addressed a major difficulty in building systems using rank-based learning)[22] . Deep learning models do not take raw text as input; they only work with numerical tensors by vectorizing text into numerical tensors. [23]

In the literature, several models of neural networks applied to IR are used in combination to achieve overall efficiency. Table 4 presents an overview of the results obtained. The citation in each row indicates the original paper where the method is proposed. The backslash symbols indicate no published results for the specific model on the specific dataset in the related literature. Based on the reported results, we observe that:

Table 4. Overview of previously published results on ad-hoc IR datasets.

	Robuste04		**ClueWeb-09-Cat-B**	
	CARD	P@20	CARD	P@20
BM25 [24]	0.255	0.370	0.101	0.326
QL [25]	0.253	0.369	0.100	0.328
DSSM [26]	0.095	0.171	0.039	0.131
CDSSM [27]	0.067	0.125	0.054	0.177
ARC-I [28]	0.041	0.065	0.024	0.089
ARC-II [28]	0.067	0.128	0.033	0.123
MP [29]	0.189	0.290	0.066	0.158
DRMM [25]	0.279	0.382	0.113	0.365
PACRR [30]	0.254	0.363	/	/
NPRF-KNRM [31]	0.285	0.393	/	/
NPRF-DRMM [31]	0.290	0.406	/	/
BERTBase MaxP[22]	0.365	0.465	/	/
BERTLarge MaxP[22]	0.374	0.477	/	/
BERT-QE-Large[22]	0.386	0.489	/	/
BERT-QE-Medium[22]	0.383	0.489	/	/
PARADE[22]	0.380	/	/	/

- Although the probabilistic models (QL and BM25) are simple. They can already achieve reasonably good performance.
- The asymmetric, interaction-oriented, multi-granularity architecture can perform better than the symmetric, representation-oriented, single-granularity architecture on ad-hoc search tasks, except for SNRM.
- As a specific instance of transformer architectures, BERT provides superior results to what has gone before. It is a robust empirical result that is widely replicated. BERT stands out for bringing together many crucial ingredients to produce tremendous advances in inefficiency. As a more sophisticated model, it draws many vital insights: the goal of contextual integrations is to capture complex features of language (e.g., syntax and semantics), as well as the way meanings vary across linguistic contexts (e.g., polysemy).

8. DISCUSSION

The IR models analyzed in this work have several contributions and suffer from several limitations:

A model based on the bag-of-words representation is simple, efficient, and easy to implement, which makes it ideal for forming the basis of more complex algorithms and IRS [32]. However, the bag-of-words representation and TF-IDF are constrained by several challenges:

- The TF-IDF treats words according to their morphology. For example, “year” and “years” will be considered as two separate words, which leads to a decrease in the weight of the word in the collection.
- An ample term space slows down the search and consumes memory space.
- The effectiveness of TF-IDF decreases with increasing collection scale[8][33]

Based on a bag-of-words representation, this approach does not consider the relationship between words. A word can have several synonyms [34]. Moreover, TF-IDF is limited to a lexical function and does not allow checking text semantics. So, it is not practical to search for co-occurrences of a word [6][8].

A graph-based model has the advantage of linking multiple and various objects with various fast and scalable techniques. Its structure is flexible to incorporate many performance improvement techniques. The results show that including term relationships and term importance weighting is helpful for search. However, incorporating automatic query expansion into the model is not very useful. Nevertheless, with the crazy evolution of the documents to be indexed before processing and the dynamic nature of the collection, the indexing space becomes larger and larger. In addition, the high frequency of use requires asking questions about the response time, the adaptability of a model, and the availability of relevant information at the right time. We mean by the adaptability of a model its ability to progressively update the indexing of the database with the constant operation when the database is periodically updated with new information[35].

A model based on classification techniques can remedy its limitations and has several advantages:

- Accelerated search process:
  number of classes < number of documents
- Adaptability and extensibility for dynamic collections to find relevant information in response to an information need, during a mass of constantly changing data.

On the other hand, the techniques treated in the present synthesis are based on statistical approaches using TF-IDF. It is then limited to a lexical function of words without considering the problems related to ambiguities. Hence the importance of observing the problem of the semantic gap and the possibilities of enriching these models with solutions that allow for the semantic aspect.

When faced with such a problem to study the meaning of a word ideally, it should be observed in its context. Thus, in a text (and by extension in language), there is more or less significant dependence between words.

A recent family of techniques (circa 2013) has rethought models with a representation of words in a space with some similarity between them (i.e., probabilistic), in which the meanings of words bring them closer together in that space in terms of statistical distances. It is folded in a dimension space. Its pet name: is word2vec or a language model in a more general way.

Language models, introduced in 1998 by Ponte and Croft [36] are a probabilistic concept. What is the probability that a query (sequence of terms (or grams)) will appear in a given document? The difficulty is establishing this model for small documents. In order to solve the lack of information during the construction of the model, different smoothing methods (Hiemstra, 98) (Song et al., 99) and different models (Berger et al., 1999) have been proposed. The results obtained by these models have shown equivalent or even better performances than the classical models. However, the term mismatch problem occurs frequently in IR. It can occur when the query is short and/or ambiguous but also in specialized domains where non-specialists make queries and documents are written by experts. Recently, the term mismatch problem has been addressed using neural learning to rank models and word plunging to avoid using only exact term matches for search. Another approach to the term mismatch problem is to use Knowledge Bases that can associate different terms with the same concept. In addition, the recent success of transformers in automatic natural language processing (NLP) [37][38][39][40], which have managed to achieve significant performance gains through sequence representations useful for this field of application. However, applying these encoders, for IR, in an architecture adopting deep learning does not lead to the same performance gain. The consulted literature presents the following explanations:

- The semantic matching used for NLP differs from the relevance matching adopted for IR. These differences affect the design of deep model architectures, and it can be difficult to find a "one-size-fits-all" solution to such different matching problems. [25][28]
- The query is usually short and based on keywords. The document can vary considerably in length, from tens of words to thousands or even tens of thousands of words.
- Deep learning techniques have been widely criticized as a "black box" that produces good results but no insights and explanations of problems.

We are nearing the end of this study, but we are still far from the end of the road in this line of research; there are still many open questions, unexplored directions, and much more work to do.

9. CONCLUSION

In this paper, we have presented some approaches to TIR. Five textual IR approaches have been discussed, those based on keywords, those based on graphs, those based on classification techniques, and those based on deep learning techniques. The traditional approaches suffer from polysemy and synonymy, while the language model-based approaches are more efficient because they allow for term linking (word-word). In addition, approaches based on ML techniques, in general, can encompass different approaches (statistical, graph, language...) on the one hand and benefit from several advantages, namely: feature extraction, system abstraction, response time, scalability, adaptability, etc. Hence the importance of investing in the advantages of the proposed approaches to have a more optimal model. The combination of several approaches is excellent potential to improve the performance of the IRS.

In our future work, we will propose a hybrid approach to IR based on the combination of several approaches of IR and deep learning techniques.

10. REFERENCES

[1] N. J. Belkin, “Helping people find what they don’t know: recommendation systems help users find the corret words for a successful search”, Communications of the ACM, Vol. 43, No. 8, 2000, pp. 58-61.

[2] B. Kitchenham, S. Charters, “Методи за автоматично управление на подемни устройства при Jack-up системите”, 2007.

[3] G. Salton, “Search and Retrieval Experiments in Real-Time Information Retrieval”, Technical Report, Cornell University, 1968, p. 16.

[4] B. A. Kitchenham, “Systematic review in software engineering: where we are and where we should be going”, Proceedings of the 2nd international workshop on Evidential assessment of software technologies, 2012, pp. 1-2.

[5] H. Benhar, A. Idri, J. L Fernández-Alemán, "Data preprocessing for heart disease classification: A systematic literature review", Computer Methods and Programs in Biomedicine, Vol. 195, 2020.

[6] S. Qaiser and R. Ali, “Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents”, International Journal of Computer Applications, Vol. 181, No. 1, 2018, pp. 25-29.

[7] G. Salton, J. Allan, C. Buckley, "Approaches to passage retrieval in full text information systems", Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 49-58, 1993.

[8] J. Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries", Proceedings of the First International Conference on Machine Learning, 2003, pp. 29-48.

[9] O. A. S. Ibrahim and D. Landa-Silva, "Term frequency with average term occurrences for textual information retrieval", Soft Computing, Vol. 20, No. 8, 2016, pp. 3045-3061.

[10] R. B.-Y. WB Frakes, "Information retrieval: data structures and algorithms", Pearson College Div, 1992.

[11] M. Gamon, "Graph-based text representation for novelty detection", Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing, June 2020, pp. 17-24.

[12] R. Blanco, C. Lioma, "Graph-based term weighting for information retrieval", Information Retrieval Journal, Vol. 15, No. 1, 2012, pp. 54-92.

[13] M. R. Islam, M. R. Islam, "An improved keyword extraction method using graph based random walk model", Proceedings of the 11th International Conference on Computer and Information Technology, 2008, pp. 225-229.

[14] O. Sornil, "A Graph-Based Information Retrieval Model", Proceedings of the Conférence en Recherche d'Information et Applications, 2008.

[15] H. Patil, R. S. Thakur, "Document Clustering", Information Retrieval and Management: Concepts, Methodologies, Tools, and Applications, 2016, pp. 264-281.

[16] C. Bouveyron, S. Girard, "Classification supervisée et non supervisée des données de grande dimension", La revue MODULAD, Vol. 40, 2009, pp. 81-102.

[17] W. Lam, M. Ruiz, P. Srinivasan, "Automatic text categorization and its application to text retrieval", IEEE Transactions on Knowledge and Data Engineering, Vol. 11, No. 6, 1999, pp. 865-879.

[18] B. Trstenjak, S. Mikac, D. Donko, "KNN with TF-IDF based framework for text categorization", Procedia Engineering, Vol. 69, 2014, pp. 1356-1364.

[19] A. Desai, C. Shrihari, "A Review on Knowledge Discovery using Text Classification Techniques in Text Mining", International Journal of Computer Applications, Vol. 111, No. 6, pp. 12-15, 2015.

[20] J. Han, M. Kamber, J. Pei, "Data Mining Concepts and Techniques", Journal of Chemical Information and Modeling, Vol. 53, No. 9, 2012, pp. 1689-1699.

[21] S. S. Deeksha, "Finding similarity in articles using various clustering techniques", Proceedings of the 6th International Conference on Reliability in Information Communication Technology, Vol. 2018, June 2018, pp. 343-347.

[22] J. Lin, R. Nogueira, A. Yates, D. R. Cheriton, C. Science, "Pretrained Transformers for Text Ranking BERT and Beyond", Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 1-204.

[23] F. Chollet, "Deep Learning with Python", Manning, 2017.

[24] S. E. Robertson, S. Walker, "Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval", Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, 1994.

[25] J. Guo, Y. Fan, Q. Ai, W. B. Croft, "A deep relevance matching model for Ad-hoc retrieval", Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, 24-28-October 2016, pp. 55-64.

[26] P. Huang et al. "Learning Deep Structured Semantic Models for Web Search using Clickthrough Data", Proceedings of the 22$_{nd}$ ACM international conference on Information & Knowledge Management, 2013, pp. 2333-2338.

[27] Y. Shen, X. He, J. Gao, L. Deng, G. Mesnil, "A latent semantic model with convolutional-pooling structure for information retrieval", Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, November 2014, pp. 101-110.

[28] J. Guo, Y. Fan, L. Pang, L. Yang, Q. Ai, "A Deep Look into neural ranking models for information retrieval", Information Processing & Management, No. June, 2019, p. 102067.

[29] H. Palangi et al. "Deep Sentence embedding using long short-term memory networks: Analysis and application to information retrieval", IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 24, No. 4, pp. 694-707, 2016.

[30] K. Hui, A. Yates, K. Berberich, G. de Melo, "PACRR: A position-aware neural IR model for relevance matching", Proceedings of the Conference on Empirical Methods in Natural Language Processing, July 2017, pp. 1049-1058.

[31] A. Yates, L. Sun, J. Xu, "NPRF: A Neural Pseudo Relevance Feedback Framework for Ad-hoc Information Retrieval", Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2018, pp. 4482-4491.

[32] A. Berger, R. Caruana, D. Cohn, D. Freitag, V. Mittal, "Bridging the lexical chasm: statistical approaches to answer-finding", Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, 2000, pp. 192-199.

[33] B. Trstenjak, S. Mikac, D. Donko, "KNN with TF-IDF based framework for text categorization", Procedia Engineering, Vol. 69, 2014, pp. 1356-1364.

[34] D. Beeferman, A. Berger, J. Lafferty, "Statistical Models for Text Segmentation", Machine Learning, Vol. 210, 1999, pp. 177-210.

[35] Z. Zhang, Z. Guo, C. Faloutsos, E. P. Xing, J. Y. Pan, "On the scalability and adaptability for multimodal retrieval and annotation", Proceedings of the 14th International Conference of Image Analysis and Processing - Workshops, Modena, Italy, 10-13 September 2007, pp. 39-44.

[36] J. M. Ponte, W. B. Croft, "A Language Modeling Approach to Information Retrieval", Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, 1998, pp. 275-281.

[37] M. C. Kenton, L. Kristina, J. Devlin, "BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding", Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 1953, pp. 4171-4186.

[38] G. Lample, A. Conneau, "Cross-lingual Language Model Pretraining", Proceedings of the Advances in Neural Information Processing Systems 32, 2018.

[39] A. Vaswani, "Attention Is All You Need", Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, 2017.

[40] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, "Language Models are Unsupervised Multitask Learners", OpenAI, Whitepaper, 2018.