COMPARATIVE ANALYSIS OF RANKING FUNCTIONS FOR RETRIEVING INFORMATION FROM MEDICAL REPOSITORY

***Narina Thakur*[1], *Deepti Mehrotra*[2], *Abhay Bansal*[3], *Manju Bala*[4]**

[1,3]Department of Computer Science & Engineering,
Amity School of Engineering Technology, Amity University, Noida, India
[2]Department of Information Technology,
Amity School of Engineering Technology, Amity University, Noida, India
[4]Department of Computer Science,
IP College for Women, 31 Shamnath Marg, Civil Lines, Delhi

Email: narinat@gmail.com[1], dmehrotra@amity.edu[2], abhaybansal@hotmail.com[3], manjugpm@gmail.com[4]

DOI: https://doi.org/10.22452/mjcs.vol32no1.2

ABSTRACT

Current and forthcoming Information Retrieval algorithms demand high mean average precision with contemporary high recall rates in the technical literature. Nevertheless, the existing state-of-the-art is still not optimized for speed, average query latency, and performance. The previous researchers presented various information retrieval models in the literature but the user search led to a ranking of documents that were hopeful to be relevant. In this paper, an evaluation of various information retrieval models is presented with a range of algorithms. The aim is to elaborate and review the current information retrieval function in the context of enterprise domain- specific search. Experiments were conducted on the OHSUMED benchmark data set from MEDLINE, a medical information database. The experimental results demonstrate that BM25F ranking function outperforms other extensively used ranking functions such as BM25, TFIDF, and Cosine on precision and recall measures.

Keywords: Information retrieval; Ranking functions; Similarity Measures; TF_IDF; BM25; COSINE; BM25F; Precision.

1.0 INTRODUCTION

Information Retrieval (IR) system obtain and stores conception-based information, e.g., taking contact number from a business card, typing the number in the cell phone to make a call, then storing the same for later use, is a form of IR. Gerry Soldier, a Computer Science (CS) Professor in Cornell [1] is the father of IR who has developed the first IR system called SMART IR system. The vast amount of data is available today, and finding information that is both relevant and comprehensive for the user is a huge challenge. IR has been an area of tremendous research and development since its first application in Libraries in the 1950s; [2] search by content on internal or external fields and databases. There has been considerable progress and success in developing strategies in Indexing, String Matching Algorithm, Text Classification, Text Clustering, IR models and Ranking. IR is to find the relevant documents to a given query. A query is a way to express the requisite information. The significant difficulties with the existing IR system are: queries entered by users are lack of description of a user need; the difference in opinion among users regarding the context of keywords; ambiguous content representation, and insufficient, inaccurate, and inadequate tool for estimation user-dependent relevance. An efficient IR ought to understand the user information need instead of query and give a response in an appraised time. IR applications differ based on a scale of Web, Personal, and Enterprise domain specific search. Web search is searching and extracting material over billions of documents stored in the computers. Personalized search is categorizing the emails sent to a company's email address for various departments' such as HR, Sales, Finance or Heads of the departments; email text analysis, or the subject field can contain some relevance to some department. Enterprise document specific search is to retrieve information or documents from the collection, such as research articles, scientific literature or internet documents. The information can be of various types such as dynamic data like the email categorization or static data like the categorization of the corpus into several domains. Similarity measures and scoring functions in IR is an active area of research that segregates documents into relevant and irrelevant.

A coefficient of similarity represents a similarity between documents, query and document, two queries or one query or batched queries. The order of presumed importance is considered to rank the documents. This paper presents a comparative analysis for finding the most relevant document for the given set of keywords by using various similarity coefficients viz. TF_IDF, Cosine, BM25, and BM25F. The paper is organized as follows: Section 2.0 and 3.0 describe the Future Research dimensions in Information Retrieval and Background. The Experiment and Results are discussed in Section 4.0, followed by a Conclusion in Section 5.0.

2.0 Future Research Dimensions in Information Retrieval

Information retrieval (IR) is about finding documents of unstructured text from a corpus based on user need [3]. IR system uses a query to extract relevant documents from the corpus. Indexing and retrieval models, i.e., how user queries the data are the critical aspects of designing an IR system. Designing an efficient IR algorithm is a challenge due to the lack of efficient, relevant ranking and retrieval techniques. Relevance is a complex notion and necessitates deliberation of many factors like topical, user relevance, binary, multi-valued relevance. An efficient IR system should understand the user information need and give a relevant response within the estimated time. IR in scientific literature suffers from the problem of obtaining high recall results rather than precise results [4] due to the massive amount of available information. The user query is a poor description of the actual information need. A user found the task of query writing as a difficult task. The keywords are sometimes too specific, therefore, no relevant documents can be found or a large number of documents extracted. Moreover, it is tough to get the precise point between two extremes Over-constrained query and Under-constrained query. Not all documents that are retrieved can be equally relevant even if the right keywords are selected; prioritization is therefore essential. Search evaluation is user-centered, and users are sometimes uncertain and ignorant of the content they are looking. A user Interaction [5] or user Feedback [6] [7] can enable the IR system to identify the content. The query is essential in understanding user intent. Various query refinement approaches are suggested in the literature to solve the problem of finding the exact keyword required for searching. The query refinement approaches are query expansion, query reformulation [8] [9], query suggestion, and relevant feedback [10]. Marc Sloan et al., [11] proposes an algorithm for similarity approximation and query reformulation by exhausting term based, query logs, user interaction and click training data. The task of an efficient searching and indexing has received considerable attention in IR literature. However, the scalability, i.e., growing data, freshness, and adaptability (Tuning for application) are still lacking in corpus/ repositories. Moreira, Catarina [12] proposed Comb SUM and CombMNZ unsupervised rank aggregation algorithms by joining multiple estimators of expertise, derived from the textual contents represented in the form of a graph of experts community citation pattern and experts profile information. An IR technique depends on a similarity measure. A measure of similarity is a subjective quantitative resemblance, which requires an objective measure known as distance. The distance function, i.e., is measured indirectly The similarity decreases as the distance increases. Similarity metrics are a set of abstractions to define to what extent the documents are similar. These documents are fundamentally similar in their content and context for the judgment / relevancy of retrieved documents to the degree. It is finding the rank based on the similarity between the documents, for example by computing the distance between two document vectors; we can determine the similarity between them. Commonly used similarity measures are Numerical, Boolean, String, Word, and concept or Semantic measure of similarity [13] [14]. The various Numerical similarity measures are:

Euclidean distance or L2 function is the most commonly used similarity measures in any distance-based algorithm [15] [16]. It is usually also termed as the sum of squared distances L2. The Euclidean distance is defined in (1).

$$\sqrt{(\sum_{i=1}^{n}(u_i - v_i)^2)}\ , \qquad (1)$$

where $i = 1...n$.

Manhattan Distance L1 is the sum of absolute distance. It is the primary similarity measure as shown in (2).

$$\sum_{i=1}^{n} |u_i - v_i\ |, \qquad (2)$$

where $i = 1 ... n$.

Chebyshev Distance and Minkowski Distance numerical similarity measures are as defined in (3) and (4) respectively.

$$\max_i |u_i - v_i\ |, \qquad (3)$$

where $i = 1 ... n$.

$(\sum_{i=1}^{n} |u_i - v_i|)^{\frac{1}{p}}$, (4)
where $i = 1 \ldots n$.

The Boolean similarity measures are Jaccard Dissimilarity, Dice Dissimilarity, Canberra Distance and their formulas as presented in (5) and (6).

Cosine Distance: $1 - u.v/(||u|||v||)$ (5)

Correlation Distance:
$1 - (u - \text{Mean}[u]).(v - \text{Mean}[v])/ (\text{Abs}\ [u - \text{Mean}[u]]\ \text{Abs}\ [v - \text{Mean}[v]])$ (6)

The String based similarity algorithms are Hamming distance, Edit Distance and Damerau Levenshtein Distance. Hamming distance is a similarity measure used for categorical attributes. Levenshtein Distance is a pairwise string alignment based string similarity measure. It is an edit distance string metric proposed in 1965 by Vladimir Levenshtein [17]. In Levenshtein Distance, the minimum number of edits with a single character is calculated using certain operations, such as insertions, deletions or substitutions, to change one word into another. For example, apples are more similar to oranges than to pears as shown in Fig. 1.

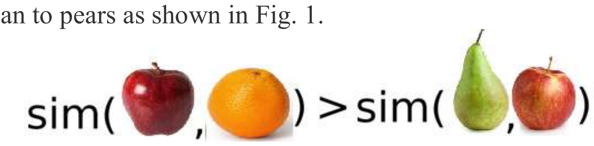


Fig. 1: Demonstrating the similarity of oranges is more to apples than to pears

Nevertheless, the process of articulating documents as similar using computer code is not that straightforward. The Cosine similarity measure is an extensively used similarity measure, but the statistical similarity measure performs better. The problem of finding the right similarity measure for the distance can be resolved by maximizing the probability of similarity. Various probability similarity measures are Kullback–Leibler [18], K-divergence, Pearson χ2, Divergence, Clark, Jensen difference and Jensen-Shannon. Other recent and upcoming similarity measures are Semantic matching [19], Graph-based and content similarity. Semantic matching is the evaluation of the similarity amongst the concepts in target ontology, connected to form concept mapping. Troels [20] describes measuring similarity in a Content-based Information Retrieval (CBIR) framework by using the similarity graph [21] which uses the fundamental similarity of native neighborhoods for the nodes of different ontologies. The various evolving research areas and approaches are content-based approaches, Author – relevancy techniques and usage ranking techniques.

3.0 BACKGROUND

The retrieval and ranking function uses the computation of a term-weighting scheme such as term frequency, inverse document frequency, document length, and normalization. The primary focus of an IR system is the speed of the search rather than the relevance of the search outcomes. Relevance is the core of the search engine, where it should be measured in the search perspective to meet the user's query needs. If document length normalization is not applied, then the long document would be ranked the short documents as the long document would have more chances of the term as compared to the short document. Document length normalization is the number of occurrences of the term divided by the document length.

This section presents the TF_IDF, Cosine, BM25 and BM25F IR algorithms. TF_IDF has been the most frequently used Term weighting algorithms where it assigns a high weight to a term if it frequently occurs in a document but rarely in the entire corpus. If the frequency of a term is high within a small number of the document collection, less likely, it will lend high discriminating power to those documents and the more the frequency of the term within the document, the more information it will carry within the document. This model is theoretically easy to understand and implement.

The TF_IDF weighting formula is shown in (7) and (8):

$$w\ (t,d) = tf\ (t,d).IDF\ (t) \quad (7)$$

$$w(t,d) = tf(t,d)\,.\ log\left({}^{N}\!/_{df}\right), \tag{8}$$

where
df: Document frequency
N: # documents in the corpus
tf (t, d) =frequency count of the term 't' in a document / total #terms in the document
w(t, d) is the weight of d document for term t; tf is the term frequency
IDF (t) = log (Total #documents in the collection / # documents where t terms appears).

Cosine similarity between two documents measures how similar two documents are in their subject matter. Cosine similarity is preferred over other similarity measures as it is independent of the vector magnitudes. It is an effective algorithm, especially for a sparse vector, which measures the angles between the vectors to calculate the similarity score. It is an orientation judgment rather than the magnitude of the two vectors. The vectors with same the directions have a cosine similarity one and zero when perpendicular and (-1) when both are opposite. Cosine similarity is not a proper distance metric [23] as it does not satisfy the triangle inequality property and it violates the coincidence axiom. It is similar to the Pearson correlation coefficient. The cosine similarity measure can be used to evade the bias caused by different document lengths. It is the inner product of the two vectors divided by the product of vector lengths. The angle cosine between the document and Query vectors is considered, and the unit length normalized vectors are used in (9).

$$Cosine\ Sim\ (Q, D_i) = \frac{\sum_{j=1}^{n} wqj * wij}{\sqrt{\sum_{j=1}^{n}(wqj)^2 * \sum_{j=1}^{n}(wij)^2}} \tag{9}$$

Okapi BM25 similarity function or BM25 is a commonly used IR ranking function due to its consistently high retrieval accuracy. BM25 performs better than TF-IDF with short documents collection [24]. BM25 uses probabilistic retrieval model developed in 1970 and 1980 by Stephen E Robertson Karen, Jones at TREC-3 in OKAPI system. The BM25 model has evolved from the BM approximations to the 2-Poisson model [25]. The formula for BM25 is as shown in (10).

$$BM25 = \sum_{t\in q} Log\left(\frac{N - n_t}{n_t}\right).\frac{(k_1+1)fd,t}{K+fd,t}\ .\frac{(k_3+1)fd,q}{k_3+fd,q} \tag{10}$$

where
q: a query,
N: # documents in the collection.
n_t*: Total # documents that contain term t.*
fd,t*: # occurrence of term t in document d i.e. tf, of term t in the current document.*
fd,q*: # occurrence of term t in document d i.e. tf, of term t in the q query.*

$$k = k_1.\left((1-b) + \frac{b.dld}{avl}\right)$$

dld: # terms in document d
avl: Average document length
$k_{1,}k_3, and$ *b are tuning parameter with values as* $k_1 = 1.2, b = 0.75, and\ k_3$*=1000*

$$score(d,q) = \sum_{t\in q} tf(t,d)\,.\,idf(t).boost(t,d).norm(d) \tag{11}$$

The normalized similarity score used for document *d* with the search term and tf-idf weight with *q* query is as shown in (9).

where
$tf(t\ in\ d)$*: Term frequency for the term t in document d as shown in (12).*

$$tf(t,d) = \frac{log(1+freq(t,d))}{log\left(1+avg\left(freq(t,d)\right)\right)} \qquad (12)$$

where
avg (freq (t, d)) is the average of freq (t, d)
idf(t): inverse document frequency of the term as shown in (13).

$$idf(t) = 1 + log\frac{numDocs}{docFreq(t)+1} \qquad (13)$$

Boost (*t. field, d*): boosting factor of the field of term t in d document and assigned during indexing.
Norm (*d*): normalized value as shown in (14).

$$norm(d) = \sqrt{0.8\ avg(\#unique\ terms) + 0.2\ \#\ uniqueTerms(d)} \ (14)$$

The Tuning parameters enable us to control the length normalization thereby improved retrieval results. The optimal value of these parameters can be determined for the test corpus using documents, queries, and judgments and optimize effective retrieval metric like Mean Average Precision (MAP). BM25F is a variant of BM25, which considers document structure and anchor text into account [26]. The formulae are as shown in (15).

$$BM25F = \sum_{t\in q\cap d}\frac{tf(t,d)}{k_1+tf(t,d)}.idf(t) \qquad (15)$$

$$Tf(t,d) = \sum_{c\in d} w_c\ .tf_c\,(t,d) \qquad (16)$$

where
c: the field contained in document d
w_c : *Weight/boost factor for each field in the document*
$tf_c(t,d)$: $Field\ tf\ function\ of\ t\ in\ the\ field\ c.$

BM25F is more suitable for structured documents while BM25 is more suitable for the unstructured documents. BM25 also perform better with short queries than the very long queries [27]. The retrieval and ranking of documents can be evaluated based on the similarity between a pair of documents. Similarity models differ based on the response selection process, which can be either Deterministic or Probabilistic if the same results obtained for each run for a randomly selected input using sampling the probability distribution. The comparison of several state-of-the-art IR models on Retrieval model, Indexing, Matching, Query type, result criteria and ordering IR criteria are presented in Table 1. Table 2 compares the various IR models on query representation and similarity operators while Table 3 highlights the strengths and weaknesses of the IR models.

Table 1: Comparison of Information Retrieval models based on various Information Retrieval Criteria

IR Criteria / IR Retrieval Model	Boolean Model	Boolean Model variants	Extended / Soft Boolean Model	Vector Space Model	Probabilistic Model
Information Retrieval Model	Deterministic	Deterministic	Deterministic	Deterministic	Probabilistic
Indexing	Complete items	Complete items	Complete items	Complete items	Derived from the content
Matching Retrieval	Exact Match	Exact Match	Exact Match	Partial or Best Match	Partial or Best Match
Query type	Structural	Structural	Structural	Structural	Natural Language
Result criteria	Any Match	Any Match	Any Match	Relevance	Relevance
Result ordering	Arbitrary	Arbitrary	Ranked	Ranked	Ranked

Table 2: Comparison of Information Retrieval Models

Model	Descriptions
Boolean Model	**Query representation:** Boolean combination of terms. **Similarity operators:** Boolean Algebra AND, OR and NOT t1 OR t2 OR ….OR (ti AND tf) OR (t1 AND tm AND tn) …. OR tl Where tl, tm, and tn are the document and query terms, which exceed the threshold.
Boolean Model variants	**Query representation:** A query is searched in the Syntactic document components rather than the whole document as: 1) Title, Abstract. 2) Specific position with e.g. word at the title beginning. 3) Proximity operators: how close in the next two terms must be to satisfy the query condition for specific units, e.g., words, sentences, paragraph order [6]. **Similarity operators:** Proximity operator and Boolean Algebra AND, OR and NOT t1 OR t2 OR …. OR (ti AND tf) OR (t1 AND tm AND tn) …. or tl Where tl, tm, and tn are the document and query terms, which exceed the threshold.
Extended /Soft Boolean Model	**Query representation:** Weights are assigned, to evaluate the output argument in the range of 0 to 1. **Similarity operators:** Extended / Soft Boolean Model Pnorm proposed by Salton [3-7] based on similarity correlation formulae ORED with Anick Approaches where, $\mathrm{SIMAND}\left(\mathrm{d}\left(\mathrm{t}_1, \mathrm{w}_{\mathrm{q}1}\right)\right)$ $\mathrm{AND}\ldots\mathrm{AND}\left(\mathrm{t}_\mathrm{n}, \mathrm{w}_{\mathrm{qn}}\right) = 1 - \left(\frac{\sum_{i=1}^{n}((1-w_{dt})^{p}.Wq_{i}{}^{p})}{\sum_{i=1}^{n} w_{qi}{}^{p}}\right)^{1/p}$ $\mathrm{SIMOR}\left(\mathrm{d}\left(\mathrm{t}_1, \mathrm{w}_{\mathrm{q}1}\right)\right) \mathrm{OR}\ldots\mathrm{OR}\left(\mathrm{t}_\mathrm{n}, \mathrm{w}_{\mathrm{qn}}\right) = \left(\frac{\sum_{i=1}^{n}(w_{di}{}^{p}.Wq_{i}{}^{p})}{\sum_{i=1}^{n} w_{qi}{}^{p}}\right)^{1/p}$

Vector Space Model	**Query representation:** Representation of query and documents: Vector of Terms Weighting Model scheme: TF_IDF (weights in the range of 0 to 1). **Similarity operators:** Cosine Similarity for TF_IDF documents. Weight: $w_{ij} = tf_{ij} \,.\, log \frac{N}{n_i}$ where, w_{ij} : The weight assigned to term i in document *j*. tf_{ij} : # occurrences of the term I in document *j*. $tf_{ij} = \frac{f_{ij}}{max\{f_{1j}\,, f_{2j,..........f_{nj}}\}}$ *N*: # documents in the entire collection. n_i : # documents with term *i*. $$Sim(d_j, q) = \frac{d_j \,.\, q}{\lVert d_j \rVert \; \lVert q \rVert} = \frac{\sum_{i=1}^{n} w_{ij}\, w_{iq}}{\sqrt{\sum_{i=1}^{n} {w_{ij}}^2}\;\sqrt{\sum_{i=1}^{n} {w_{iq}}^2}}$$
Probabilistic Model	**Query representation:** Binary vectors [2]. **Similarity operators :** Each vector element indicates whether a document term occurs in the document or not. It uses probabilistic, instead of probability where, $$O(R\,) = \, P(R) \big/ 1 - P(R)$$ Okapi BM25 similarity function : $BM25 = \sum_{t \in q} Log\left(\frac{N - n_t}{n_t}\right) . \frac{(k_1+1)fd,t}{K+fd,t} \, . \frac{(k_3+1)fd,q}{k_3+fd,q}$ where *q* is the query, *N*: # documents in the corpus. n_t:To tal # documents that contain term *t*. fd, t: # occurrence of term *t* in document d i.e. *tf,* of term *t* in the current document. fd, q: # occurrence of term *t* in document d i.e. *tf,* of term *t* in the *q* query. $k = k_1 . \left((1 - b) + \frac{b.dld}{avl} \right)$ *dld*: # terms in the document, *d*. *avl*: average document length. *k1, k3,* and *b* are a free parameters $k_1 = 1.2, b = 0.75$ and k_3 = 1000

Table 3: Strengths and Weaknesses of the Information Retrieval models

Model	Strengths	Weaknesses
Boolean Model And Boolean Model variants	The retrieved documents can be either limited or voluminous and also relevant or irrelevant.	The size of the resultset is unpredictable where it can be either too many retrieved documents or none. It considers all the retrieved documents in the resultant posting list. Since all terms are weighted equally, the retrieved documents will not be ranked. Hence, all documents are considered “equally worthy.” Documents that “don’t fairly matched” the query may be beneficial also. There is no provision for partial matches
Extended / Soft Boolean Model	It is a simple model based on Linear Algebra and Term weights. The term weights are not binary. This model allows partial document matching.	Formulating useful extended Boolean model requires more thought and expertise in the query domain.
Vector Space Model	It is a Term weight model based on a Geometric similarity measure. It uses the Dot product of the query and document vector and can even allow partial matching.	Lengthy documents have little similarity scores. Precise match of the query keywords in the document terms may result in false positive results due to substring match.
Probabilistic Model	**G**iven a query, the model ranks documents by the probability of relevance.	Independent assumption and Parameter estimation are the two crucial principal concern issues. It is hard to estimate parameters, i.e. need to estimate relevance, without a proper training dataset. In the composite terms, the presence of one term increases the likelihood of the presence of the other even though sometimes it is not realistic.

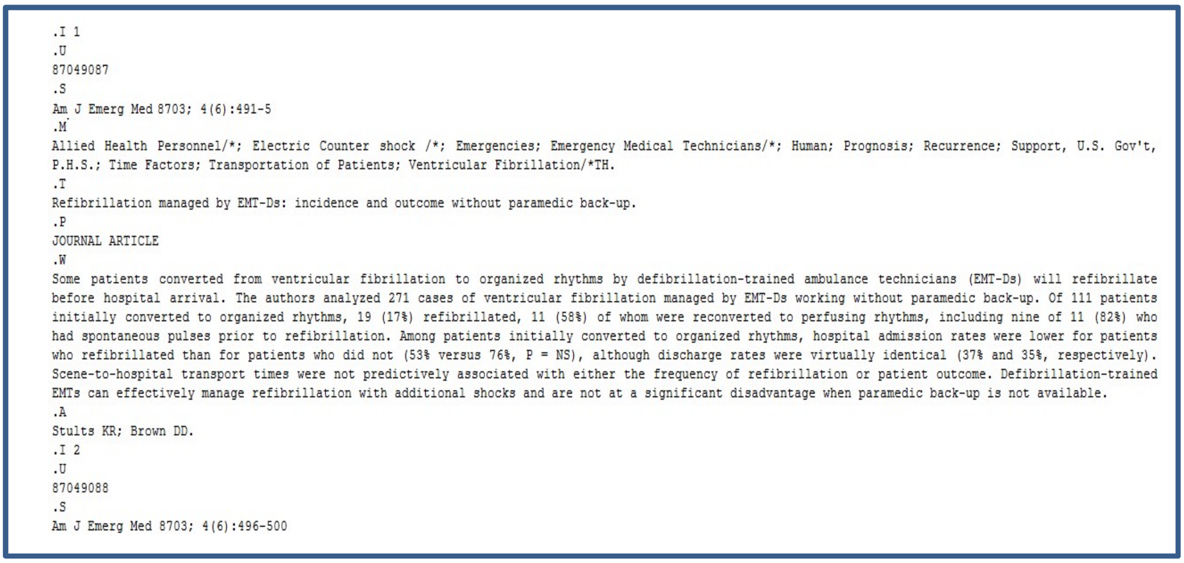
4.0 EXPERIMENTS AND RESULTS

The experiments are carried on the TREC OHSUMED dataset [28]. TREC Conference was started in 1992 and was co-sponsored by the National Institute of Standards and Technology and U.S. Department of Defense. It was also part of the TIPSTER Text program. William Hersh and his colleagues obtained OHSUMED corpus [29] for their experiments. The OHSUMED is an online medical information database, MEDLINE, which contains 348,566 references, collected over five years, and it consists of 270 medical journal titles and abstracts from 1987 to 1991. The available fields as shown in Table 4 and their definitions.

Table 4: Field definitions of the TREC-9 OHSUMED Dataset

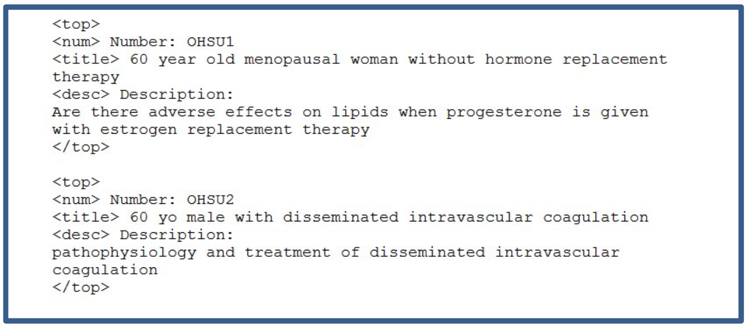
Notation	Field	Short form
.I	Sequential identifier	
.U	MEDLINE identifier, <DOCNO> used for relevance judgments	UI
.M	Human-assigned MeSH terms	MH
.T	Title	TI
.P	Publication type	PT

Fig. 2 and Fig. 3 represent the OHSUMED data set schematic view and QREL schematic view.



```
.I 1
.U
87049087
.S
Am J Emerg Med 8703; 4(6):491-5
.M
Allied Health Personnel/*; Electric Counter shock /*; Emergencies; Emergency Medical Technicians/*; Human; Prognosis; Recurrence; Support, U.S. Gov't,
P.H.S.; Time Factors; Transportation of Patients; Ventricular Fibrillation/*TH.
.T
Refibrillation managed by EMT-Ds: incidence and outcome without paramedic back-up.
.P
JOURNAL ARTICLE
.W
Some patients converted from ventricular fibrillation to organized rhythms by defibrillation-trained ambulance technicians (EMT-Ds) will refibrillate
before hospital arrival. The authors analyzed 271 cases of ventricular fibrillation managed by EMT-Ds working without paramedic back-up. Of 111 patients
initially converted to organized rhythms, 19 (17%) refibrillated, 11 (58%) of whom were reconverted to perfusing rhythms, including nine of 11 (82%) who
had spontaneous pulses prior to refibrillation. Among patients initially converted to organized rhythms, hospital admission rates were lower for patients
who refibrillated than for patients who did not (53% versus 76%, P = NS), although discharge rates were virtually identical (37% and 35%, respectively).
Scene-to-hospital transport times were not predictively associated with either the frequency of refibrillation or patient outcome. Defibrillation-trained
EMTs can effectively manage refibrillation with additional shocks and are not at a significant disadvantage when paramedic back-up is not available.
.A
Stults KR; Brown DD.
.I 2
.U
87049088
.S
Am J Emerg Med 8703; 4(6):496-500
```

Fig. 2: Schematic view of OHSUMED dataset file



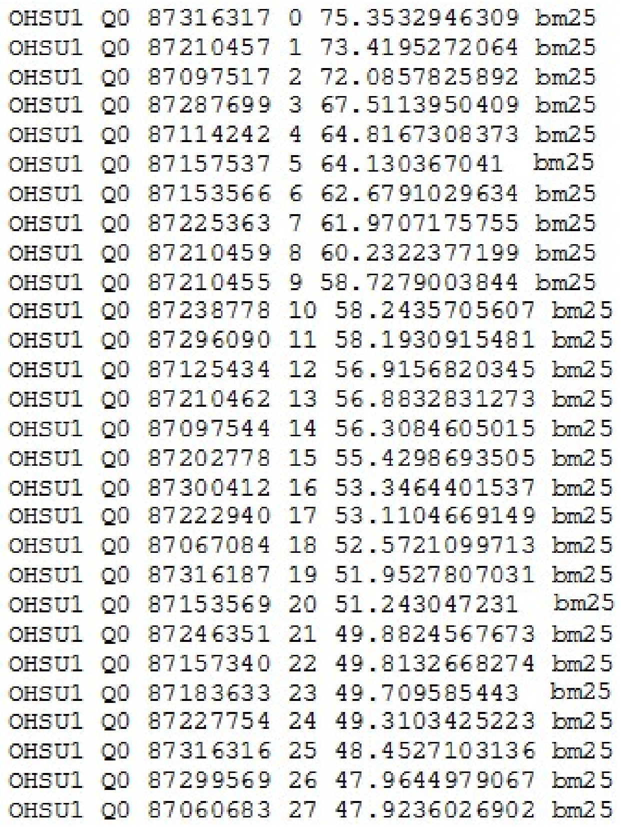
```
<top>
<num> Number: OHSU1
<title> 60 year old menopausal woman without hormone replacement
therapy
<desc> Description:
Are there adverse effects on lipids when progesterone is given
with estrogen replacement therapy
</top>

<top>
<num> Number: OHSU2
<title> 60 yo male with disseminated intravascular coagulation
<desc> Description:
pathophysiology and treatment of disseminated intravascular
coagulation
</top>
```

Fig. 3: Schematic view of OHSUMED QREL

The implementation of the IR model uses Whoosh library and Python programming for indexing and searching the OHSUMED dataset for a given set of 60 QRELS.

The various ranking models compared and analyzed are:

1. TF-IDF
2. Cosine
3. PL2
4. BM25
5. BM25F



```
OHSU1 Q0 87316317 0 75.3532946309 bm25
OHSU1 Q0 87210457 1 73.4195272064 bm25
OHSU1 Q0 87097517 2 72.0857825892 bm25
OHSU1 Q0 87287699 3 67.5113950409 bm25
OHSU1 Q0 87114242 4 64.8167308373 bm25
OHSU1 Q0 87157537 5 64.130367041  bm25
OHSU1 Q0 87153566 6 62.6791029634 bm25
OHSU1 Q0 87225363 7 61.9707175755 bm25
OHSU1 Q0 87210459 8 60.2322377199 bm25
OHSU1 Q0 87210455 9 58.7279003844 bm25
OHSU1 Q0 87238778 10 58.2435705607 bm25
OHSU1 Q0 87296090 11 58.1930915481 bm25
OHSU1 Q0 87125434 12 56.9156820345 bm25
OHSU1 Q0 87210462 13 56.8832831273 bm25
OHSU1 Q0 87097544 14 56.3084605015 bm25
OHSU1 Q0 87202778 15 55.4298693505 bm25
OHSU1 Q0 87300412 16 53.3464401537 bm25
OHSU1 Q0 87222940 17 53.1104669149 bm25
OHSU1 Q0 87067084 18 52.5721099713 bm25
OHSU1 Q0 87316187 19 51.9527807031 bm25
OHSU1 Q0 87153569 20 51.243047231  bm25
OHSU1 Q0 87246351 21 49.8824567673 bm25
OHSU1 Q0 87157340 22 49.8132668274 bm25
OHSU1 Q0 87183633 23 49.709585443  bm25
OHSU1 Q0 87227754 24 49.3103425223 bm25
OHSU1 Q0 87316316 25 48.4527103136 bm25
OHSU1 Q0 87299569 26 47.9644979067 bm25
OHSU1 Q0 87060683 27 47.9236026902 bm25
```

Fig. 4: Schematic view of Output generated from the Information Retrieval Python program for the BM25F Model

4.1 Evaluation

An empirical comparison of the performance of standard popular ranking models is presented in this paper. Experiments were conducted on the TREC-9 OHSUMED dataset. The performance of an IR system can be evaluated with standard precision and recall metrics. These measurements are used to measure accuracy in the ranking and search of documents, respectively. Precision is a measure of exactness [32], and it estimates "how well it eliminates unwanted documents," whereas Recall is a measure of completeness which measures "how well an IR system finds what user wants." The relevant documents are searched from the corpus and will be ranked according to their relevancy. Searching divides the corpus into two sets viz. a set of relevant documents that are returned from the query and a set of non-relevant which are not matched with the query [30], [31]. A similarity score for the algorithms discussed in section 3.0 is computed for the ranking or retrieval model using Python programming and Whoosh Library. The scores obtained are stored in an output file as shown in Fig. 4 . The output is then used to execute the TRECEVAL Script. TRECEVAL is the standard evaluation procedures or script from NIST which generates the output that can be used for calculating the Precision and Recall.

4.2 Discussion

The performance of all the ranking algorithms are evaluated using four common measures: Precision at 5(P@5), Precision at 10(P@10), Precision at 100(P@100), and MAP values on the TREC-9 datasets.

Table 5: Mean Average, Precision @ 5, Precision @ 10, Precision at 100 results using various algorithms on TREC-9 OHSUMED Dataset.

Algorithm/ Precision	Tf_idf	Cosine	PL2	BM25	Bm25F
P5	0.1841	0.346	0.3933	0.4159	0.4195
P@10	0.1429	0.2635	0.3524	0.3683	0.3873
P@100	0.0489	0.0594	0.1663	0.167	0.1706
P@1000	0.0049	0.0059	0.0315	0.0315	0.0317
MAP	0.14836	0.24986	0.4005	0.41422	0.4256

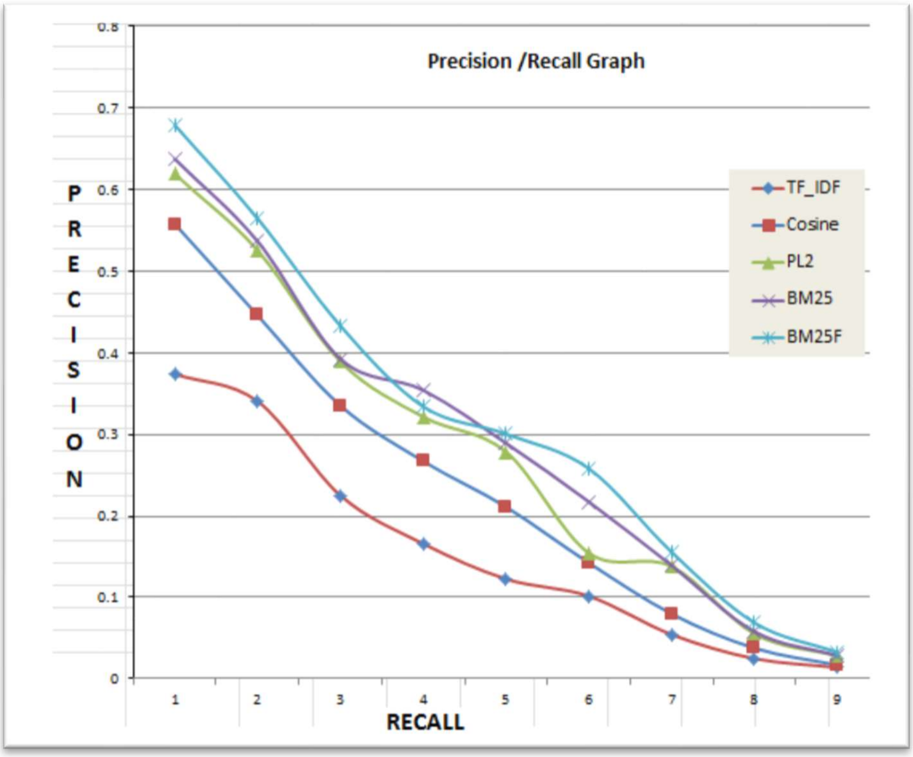


Fig. 5: Precision / Recall Graph

As shown in Table 3, the BM25F algorithm outperforms the other algorithms on the TREC-9 OHSUMED Dataset. Fig. 5 represents Precision / Recall graph which also shown that BM25F algorithm outperforms other algorithms.

5.0 CONCLUSION

In this paper, we have analyzed and empirically reviewed the various IR ranking algorithms. The Scoring and Similarity measures are evaluated to identify the relationship between the current similarity functions in the context of IR systems. In our experiment using TREC-9 OHSUMED Medical dataset, we found that Cosine and BM25 measures have comparably effective, but BM25F algorithm in generally outperforms the existing ranking measure. This paper reviews the state-of-the-art IR techniques and suggests new approaches. It provides a detailed explanation of how the current information retrieval approaches work; examine the strengths and weaknesses, and identify the gaps. Furthermore, maximization of the similarity probability, by using divergence between the document and query probability, Query refinement, content-based approaches, Author – relevancy techniques and usage ranking techniques can resolve the issue of relevancy and the demands of high MAP with existing high Recall rates in the technical literature.

REFERENCES

[1] CS Cornell University, http://www.cs.cornell.edu/gries/40brochure/pg24_25.pdf.

[2] Christopher D. Manning et al., Introduction to Information Retrieval. Cambridge University Press Cambridge, England 2009.

[3] Baeza-Yates et al., "Information retrieval in the web: beyond current search engines." International Journal of Approximate Reasoning Vol. 34, No. 2-3, November 2003, pp. 97-104.

[4] Thompson, Cynthia A. et al., "A personalized system for conversational recommendations." Journal of Artificial Intelligence Research, Vol. 21, March 2004, pp. 393-428.

[5] Dang, et al., "Fast forward index methods for pseudo-relevance feedback retrieval." ACM Transactions on Information Systems (TOIS) Vol. 33, No. 4, May 2015, pp. 19.

[6] Frank Hopfgartner et al., "Second International Workshop on Gamification for Information Retrieval," in Advances in Information Retrieval, Springer International Publishing, March 2015, pp. 838-840.

[7] Lee Shaoshing, et al., "Explicit Graphical Relevance Feedback for Scholarly Information Retrieval," in iConference 2015 Proceedings, March 2015.

[8] Ailon, Nir, "A simple linear ranking algorithm using query dependent intercept variables," in European Conference on Information Retrieval. Springer Berlin Heidelberg, April 2009, pp. 685-690

[9] Tetsuya Sakai et al.," On information retrieval metrics designed for evaluation with incomplete relevance assessments". Information Retrieval Vol. 11, No. 5, Oct. 2008, pp. 447–470.

[10] Ram Gopal Raj and Sameem Abdul-Kareem. "A Pattern Based Approach for The Derivation Of Base Forms Of Verbs From Participles And Tenses For Flexible NLP". Malaysian Journal of Computer Science, Vol. 24, No. 2, Jun 2011, pp. 63-72.

[11] C. Moreira et al., "Using rank aggregation for expert search in academic digital libraries." Simpósio de Informática (INForum'11), Jan 2015, pp. 1-10.

[12] Wolfram, https://reference.wolfram.com/language/guide/DistanceAndSimilarityMeasures.html

[13] Sung-Hyuk Cha, "Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions". International Journal of Mathematical Models and Applied Science, Vol. 1, No. 4, 2007, pp. 300-307.

[14] Mielke et al., Permutation methods: a distance function approach. Springer Science & Business Media, 2007.

[15] Huang, Anna, "Similarity measures for text document clustering," in Proceedings of the sixth new Zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand, April 2008, pp. 49-56.

[16] Yujian Li et al., "A normalized Levenshtein distance metric." IEEE transactions on pattern analysis and machine intelligence Vol. 29, no. 6, Jun 2007, pp. 1091-1095.

[17] Brigitte Bigi, "Using Kullback-Leibler Distance for Text Categorization," in European Conference on Information Retrieval ECIR 2003, LNCS 2633, Springer-Verlag Berlin Heidelberg, April 2003, pp. 305–319.

[18] Sheng Qiuyan et al., "Measuring semantic similarity in ontology and its application in information retrieval", in Image and Signal Processing, CISP'08. Congress on 2008, vol. 2, IEEE, 2008 pp. 525-529.

[19] Andreasen Troels et al., "Similarity from conceptual relations," in Fuzzy Information Processing Society,

NAFIPS 2003, 22nd International Conference of the North American, IEEE, Jul 2003, pp. 179-184.

[20] Zager Laura A. et al., "Graph similarity scoring and matching" Applied mathematics letters 21, No. 1, Jan 2008, pp. 86-94.

[21] K. Sparck Jones, "A Statistical Interpretation of term specificity and its application in retrieval." Journal of Documentation, Vol. 28, No. 1, Jan 1972, pp. 11-21.

[22] Lv Yuanhua et al., "When documents are very long, bm25 fails!", in Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, ACM, Jul 2011, pp. 1103- 1104.

[23] Krysta M. Svore, et al., "A Machine Learning Approach for Improved BM25 Retrieval". Microsoft Research Technical Report MSR-TR-2009-92 July 30, 2009, pp. 1- 25. http://research.microsoft.com/pubs/101323/LearningBM25MSRTechReport.pdf

[24] S. E. Robertson et al., "The Probabilistic Relevance Framework: BM25 and Beyond". Foundations and Trends in Information Retrieval, Vol. 3, No. 4, Dec 2009,pp. 333-389.

[25] Singhal et al., "Modern Information Retrieval: A Brief Overview." Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, Vol. 24, No. 4, Dec 2001, pp. 35–43.

[26] NIST : TREC http://trec.nist.gov/overview.html.

[27] Svore Krysta M. et al., "A machine learning approach for improved BM25 retrieval", in Proceedings of the 18th ACM conference on Information and knowledge management, CIKM'09, ACM, Nov 2009, pp. 1811-1814.

[28] Hersh William et al., "OHSUMED: An interactive retrieval evaluation and new large test collection for research", in SIGIR'94, Springer London, 1994, pp. 192-201.

[29] K. H. Brodersen et al., "The binormal assumption on precision-recall curves," in International Conference on Pattern Recognition ICPR, IEEE, Aug 2010, pp. 4263-4266.

[30] Mooers Calvin N., "Theory Digital Handling Non-numerical Information." Zator Technical Bulletin No. 48 5, cited in" information, n.". OED Online. December 2011."

[31] Doyle, Lauren B., "Information retrieval and processing." Los Angeles, CA, Melville 1975. - 425 p.

[32] Mac Mullin Susan E. et al., "Problem dimensions and information trait." The information society, an International Journal, Taylor Francis, Vol. 3, No. 1, Jan 1984, pp. 91-111.