ABSTRACT
Increased interest in developing methods that can efficiently categorize and retrieve relevant textual information through search engines on internet has been noticed among researchers. In literature we find many such retrieval modeling techniques. A comparative study of all these has been missing that can channelize the research focus. In this article we present a comparative study of various Best-Match Information Retrieval Techniques for word document.

KEYWORDS
Keyword-Base Retrieval; Best-Match Retrieval; Boolean retrieval; Vector Space Model; Hyperspace Analog to Language; Probabilistic Hyperspace Analog to Language Model; Extended Probabilistic Hyperspace Analog to Language Model.

1. INTRODUCTION
Retrieval techniques for useful information to the surfer on internet have been the interest of researchers in recent years. As mentioned therein [1], because, there exist a set of documents on a range of topics, written by different authors, at different times, and at varying levels of depth, detail, clarity, and precision, and a set of individuals who, at different times and for different reasons, search for recorded information that may be contained in some of the documents in this set. In each instance in which an individual seeks information, he or she will find some documents of the set useful and other documents not useful. How should a collection of documents be organized / indexed so that a person can find all and only the relevant items? One answer is automatic information retrieval (IR) system. The goal of IR is to find the documents relevant to a query. By relevant, we usually mean that the retrieved documents should be about the same topic as the query. This does not mean that it is a necessary and sufficient condition that a relevant document contains all the keywords of the query. For example, it is possible that a document about “doctor” may not contain word “doctor” but may have word physician or cardio, so it does not mean that, this document is not relevant to a query having a word “doctor” in the query. These problems are referred as the synonymy and polysemy problems. In the literature we find a lot many information retrieval models. There are basically two broad categories of IR model, Exact-Match IR (also known as Boolean retrieval) and Best-Match IR. Exact match IR is based on the concept of an exact match of a query specification with one or more text surrogates. The term “Boolean” is used because the query specifications are expressed as words or phrases combined using the standard operators of Boolean logic. As mentioned therein [1], in this IR model all surrogates, texts, containing the combination of words or phrases specified in the query are retrieved, and there is no distinction made between any of the retrieved documents. Thus, the result of the comparison operation in Boolean retrieval is a partition of the database into a set of retrieved documents, and a set of non-retrieved documents. A major problem with this model is that it does not allow for any form of relevance ranking of the retrieved document set [1]. To overcome the problem, Best-Match retrieval models have been proposed in response to the problem of Exact-Match retrieval.

In this paper we present various Best-Match Retrieval techniques for web document with their merits and demerits.

2. BEST-MATCH RETRIEVAL MODELS
These models treat texts and queries as vectors in a multidimensional space, the dimension of which are the words used to represent the texts. Queries and texts are compared by comparing the vectors, using some correlation function such as cosine correlation. The assumption is that the more similar a vector, the more likely that the text is relevant to that query. In these models, an important refinement is that the terms (or dimensions) of a query, or text representation can be weighted, to take account of their importance. These weights are computed on the basis of the statistical distributions of the terms in the database, and in the texts [1]. In literature we find following Best-Match IR Model:

2.1 VECTOR SPACE MODELING (VSM)
The VSM (also known as tf-idf model) is implemented by creating the term–document matrix and a vector of query. Let the list of relevant terms be numbered from 1 to m and documents be numerated from 1 to n. The term-document matrix is m x n matrix $A = [aij]$, where $aij$ represents the weight of term i in document j. On the other side, we have a query or customer’s request. In the VSM, queries are presented as m-dimensional vectors. The simple VSM is based on literal matching of terms in the documents and the queries. But we certainly know that literal matching of terms does not necessarily retrieve all relevant documents. Synonyms (more
words with the same meaning) and polysemies (words with multiple meaning) are two major obstacles in information retrieval. In literature we find following two Indexing scheme based on VSM.

2.1.1 Latent Semantic Indexing (LSI)
The basic idea of LSI in Information Retrieval (IR) was proposed in 1988 by Scott Deerwester. LSI was introduced in 1990 [2] and improved in 1995 [3]. It is unsupervised dimensional reduction technique. It tries to overcome the problems of lexical matching by using statistically derived conceptual indices instead of individual words for retrieval. It represents documents as approximations and tends to cluster documents on similar topics even if their term profiles are somewhat different. This approximate representation is accomplished by using low-rank singular value decomposition (SVD) approximation of the term-document matrix. Kolda and O’ Leary in [4] proposed replacing SVD in LSI by the semi-discrete decomposition that saves memory space. Although the LSI method has empirical success, it suffers from the lack of interpretation for the low-rank approximation and, consequently, the lack of controls for accomplishing specific tasks in information retrieval. The explanation of LSI efficiency in terms of multivariate analysis is provided in [5-8]. Unfortunately, the high computational and memory requirements of LSI and its inability to compute an effective dimensionality reduction in a supervised setting limit its applicability [9]. The founder of LSI itself make a statement that LSI model deals nicely with synonymy problem, but it offer only a partial solution to the polysemy problem [2].

2.1.2 Concept Indexing (CI)
As mentioned therein [13], CI is a fast dimensionality reduction algorithm. It is supervised as well as unsupervised dimensionality reduction technique. It can be used for both supervised and unsupervised dimensionality reduction. The key idea behind this dimensionality reduction scheme is to express each document as a function of the various concepts present in the collection. This is achieved by first finding groups of similar documents, each group potentially representing a different concept in the collection, and then using these groups to derive the axes of the reduced dimensional space. In CI dimensionality reduction algorithm, the documents are represented using VSM [10]. These techniques are primarily used for improving the retrieval performance, and to a lesser extent for document categorization. Examples of such techniques include Principal component Analysis (PCA) [26], LSI [2-3, 5-8, 14, 29], Kohonen Self-Organizing Map (SOFM) [27], and Multidimensional Scaling (MDS) [28]. In this model, each document d is considered to be a vector in the term space. In its simplest form, each document is represented by the term frequency (TF) vector \( \hat{d}_{\text{tf}} = (tf_1, tf_2, \ldots, tf_n) \), where \( tf_i \) is the frequency of the \( i^{\text{th}} \) term in the document. A widely used refinement to this model is to weight each term based on its inverse document frequency (IDF) in the document collection. The motivation behind this weighting is that terms appearing frequently in many documents have limited discrimination power, and for this reason they need to be de-emphasized. This is commonly done in [10-11] by multiplying the frequency of each term \( i \) by \( \log \left( \frac{N}{df_i} \right) \), where \( N \) is the total number of documents in the collection, and \( df_i \) is the number of documents that contain the \( i^{\text{th}} \) term (i.e. document frequency). This leads to the tf-idf representation of the document, i.e. \( \hat{d}_{\text{tf-idf}} = (tf_1 \log \left( \frac{N}{df_1} \right), tf_2 \log \left( \frac{N}{df_2} \right), \ldots, tf_n \log \left( \frac{N}{df_n} \right) \). Finally, in order to account for documents of different lengths, the length of each document vector is normalized so that it is of unit length, i.e., \( \left\| \hat{d}_{\text{tf-idf}} \right\|_2 = 1 \).

In the VSM, the similarity between two documents \( d_i \) and \( d_j \) is commonly measured using the cosine function [10], given by
\[
\cos(\hat{d}_i, \hat{d}_j) = \frac{\hat{d}_i \cdot \hat{d}_j}{\left\| \hat{d}_i \right\|_2 \left\| \hat{d}_j \right\|_2},
\]
where “\( \cdot \)” denotes the dot-product of the two vectors. Since the document vectors are of unit length, the above formula is simplified to \( \cos(\hat{d}_i, \hat{d}_j) = \hat{d}_i \cdot \hat{d}_j \).

Given a set \( S \) of documents and their corresponding vector representations, we define the centroid vector \( \hat{c} \) to be
\[
\hat{c} = \frac{1}{|S|} \sum_{d \in S} \hat{d},
\]
which is the vector obtained by averaging the weights of the various terms in the document set \( S \). So similarity between a document vector \( \hat{d} \) and a centroid vector \( \hat{c} \) is computed using the cosine measure, as follow:
\[
\cos(\hat{d}, \hat{c}) = \frac{\hat{d} \cdot \hat{c}}{\left\| \hat{d} \right\|_2 \left\| \hat{c} \right\|_2} = \frac{\hat{d} \cdot \hat{c}}{\left\| \hat{c} \right\|_2^2}.
\]
Here document vectors are of unit length, but the centroid vector will not of unit length. And this document-to-centroid similarity function tries to measure the similarity between a document and the documents belonging to the supporting set of centroid. In particular, the similarity between \( \hat{d} \) and \( \hat{c} \) is the ratio of the dot-product between \( \hat{d} \) and \( \hat{c} \), divided by the length of \( \hat{c} \). According to [12] Experiments result show that centroid based document classification algorithm consistently and substantially outperforms other algorithms such as Naive Bayesian, K-nearest-neighbors, and C4.5, on a wide range of datasets. Moreover experimental results show that CI achieves comparable retrieval performance to that obtained using LSI. And the amount of time required by CI to find the axes of the reduced dimensionality space is significantly smaller than that required by LSI. CI finds these axes by just using a fast clustering algorithm, whereas LSI needs to compute the singular-value-decomposition. Experiments results also show that CI is consistently eight to ten times faster than LSI [13].

2.2 LANGUAGE MODELING (LM)
The basic idea of LM in IR given by researchers Ponte and Craft in 1998 [15]. The motive of this model is to provide an
adequate indexing model so that integration of models of document indexing and document retrieval can be achieved [15]. LM approaches to information retrieval are attractive and promising because they connect the problem of retrieval with that of language model estimation, which has been studied extensively in other application areas such as speech recognition. Experiment results show that LM outperforms standard tf.idf weighting models [15]. The basic idea of these approaches is to estimate a language model for each document, and to then rank documents by the likelihood of the query according to the estimated language model [16]. A central issue in language model estimation is smoothing, the problem of adjusting the maximum likelihood estimator to compensate for data sparseness. In LM approach to IR, one considers the probability of a query as being generated by a probabilistic model based on a document [17]. For a query \( q = q_1, q_2…q_n \) and document \( d = d_1, d_2 …d_m \), this probability is denoted by \( \text{hyp}(q|d) \). In order to rank documents, we need to estimate probability \( p(d|q) \), which from Baye’s formula is given by

\[
p(d|q) \propto p(q|d)p(d)
\]

where \( p(d) \) is our prior belief that \( d \) is relevant to any query and \( p(q|d) \) is the query likelihood given the document, which captures how well the document “fits” the particular query \( q \). \( p(d) \) is assumed to be uniform and it can be used for nontextual information. An important operation in LM is smoothing of the document language model. The term smoothing, refers to the adjustment of the maximum likelihood estimator of a language model, so that it will be more accurate. But in LM no relationships between terms are considered and no inference is involved.

2.2.1 Inferential Language Modeling:
In traditional LM (as outlined in [18]) no relationship between terms is considered and no inference is involved. But inferential LM is capable of inference using term relationship. The inference operation is carried out through semantic smoothing either on document model or query model, resulting in document or query expansion. Experiment results show that term relationships into the language modeling framework can consistently improve the retrieval effectiveness compared with the traditional language models. Inferential Language Models have been tested on several Text REtrieval Conference (TREC) collections, both in English and Chinese. This study shows that LM is suitable framework to implement basic inference operations in IR effectively. These details of Inferential LM are available in [18].

2.2.2 Cluster-Based Language Models:
Cluster-based retrieval is based on the hypothesis that similar documents will match the same information needs. In document-based retrieval, an IR system matches the query against documents in the collection and returns a ranked list of documents to the user. This type of models has been employed in topic detection and tracking (TDT) research [19-21]. Document clustering is used to collections around topics. Each cluster is assumed to be representative of a topic. Language models estimated for clusters and are used to properly represent topics and effectively select the right topics for a given query. X. Liu and W. Bruce Croft in [30] proposed two language models for cluster-based retrieval, one for ranking/retrieving and other for using clusters to smooth documents. They evaluated these models using several TREC collections based on static or query-specific clusters. Based on experiment results, they conclude cluster-based retrieval is feasible in LM framework [30]. The detail of cluster-based language model is available in [30].

2.3 HYPERSPACE ANALOG TO LANGUAGE MODELING (HALM)
The HAL model builds a high-dimensional context space to represent words. Each word in the HAL space is denoted as a vector of its neighboring context, implying that the sense of a word can be inferred from its neighboring context [25]. It is a model of semantics which derives representations for words from analysis of text. The representations are formed by an analysis of lexical co-occurrence and can be compared to measures of word similarity. HAL Space is constructed automatically from a high dimensional semantic space over a corpus of text [22], and is defined as follows: each term \( t \) in the vocabulary \( T \) is composed of a high dimensional vector over \( T \), resulting in a \([T] \times [T] \) HAL matrix, where \( T \) is number of terms in the vocabulary. A window of length \( K \) is moved across the corpus of text at one term increments ignoring punctuation, sentence and paragraph boundaries. All terms within this window are said to co-occur with the first term in the window with strengths inversely proportional to the distance between them. The weighting assigned to each co-occurrence of terms is accumulated over the entire corpus. The HAL weighting for a term \( t \) and any other term \( t’ \) is given by:

\[
\text{HAL}(t’|t) = \sum_{k=1}^{K} w(k) n(t, k, t’)
\]

where \( n(t, k, t’) \) is number of times term \( t’ \) occurs a distance \( k \) away from \( t \), and \( w(k) = K-k+1 \) denotes the strength of relationship between the two terms given \( k \). [23-24].

- Probabilistic Hyperspace Analog to Language Modeling (pHAL): Song and Bruza [31] introduce IR based on Gardenfor’s three cognitive models, Conceptual Spaces [24, 32]. They instantiate a conceptual space using HAL [22] to generate higher order concepts which are later used for adhoc retrieval [24]. As proposed in [24] an alternative implementation of the conceptual space by using pHAL space. Experiment results in [23-24] shows that probabilistic HAL (pHAL) outperforms the original HAL method. The detail of pHAL is available in [24, 25].

- Extended Probabilistic Hyperspace Analog to Language Modeling (epHAL): epHAL is applied with close temporal association for psychiatric query document retrieval in [25]. In epHAL two primary parameters, the reliability
coefficient and combination factor were presented to improve the language model performance. According to [25] experiments result indicates that the epHAL model achieves the best dynamic reliability coefficient and dynamic combination factor performance. Rather than using dynamic reliability coefficients, static coefficients can achieve feasible performance while reducing computational complexity. Applying the proposed epHAL model to psychiatric query document retrieval outperforms conventional approaches, including VSM-based models and the pHAL model. Additionally, recall and precision can enhanced based on information flow expansion and high-order constituents. The detail of epHAL model is available in [25].

3. COMPARISON AMONG VARIOUS IR MODELS
In this paper we briefly describe various popular IR model. We describe broadly two categories of IR model, one is Exact-Match retrieval and second is Best-Match retrieval Model. In Exact-match retrieval model, exact keyword matching is carried out. This is suffering from the problem of synonymy and polysemy. Best-Match retrieval model is designed to overcome these problems. In VSM (a Best-Match retrieval technique) we present two popular techniques, one is LSI and 2nd is CI. LSI technique is based on Singular Value Decomposition (SVD) and CI technique is based on Concept Decomposition (CD). According to various tests (TEST A to TEST D), conducted in [14] shows that CI is better interpretable compared to LSI. Moreover, experiment results in Table 4 and Table 5 of [9], shows that CI dramatically improves the retrieval performance for all the different classes in each data set and outperforms LSI in all classes. Table 3 and Table 4 of [13] show that, the amount of time required by CI to find the axes of the reduced dimensionality space are significantly smaller than that required by LSI. And Table 5 of [13] show that the run-time comparison of CI is consistently eight to ten times faster than LSI.

In 1998 Ponte and Craft proposed LM [15] which outperformed VSM. Empirical results in Table 1 and Table 2 of [15] shows that on the eleven point recall/precision section, the LM approach achieves better precision at all the levels of recall, significantly at several levels. Also notice that there is a significant improvement in recall, interpolated average precision and R-precision, the precision after R documents where R is equal to the number of relevant documents for each query. In [18] a series of experiments conducted on four TREC collections- three of them are English Collections and one Chinese collection. And according to Table I and Table II of [18], this series of experiment results shows that inference implemented as document expansion (Inferential LM), can improve IR effectiveness on both English and Chinese documents regardless of the language. Empirical results in Table 1 to table 5 of [30] shows that cluster-based retrieval in LM has performed significantly better than document based retrieval in the context of query likelihood retrieval.

According to Experiment-1 to Experiment-4 in [22] shows that HAL focused on word meaning (semantic of word) which outperformed Latent Semantic Analysis (LSA) [33]. Moreover according to table VI of [25], experiment result shows that HAL-based models achieved much higher precision than VSM-based models. In [22] it has been argued that HAL’s contextually-derived representations can provide sources of information that may be useful to higher-level systems and presented simulation evidence that HAL’s vector representations can provide sufficient information to make semantic, grammatical, and abstract distinctions. According to table 1 of [24] experiment result shows that pHAL-based models achieved much higher precision than original HAL-based models. And according to Table VI of [25] experiment result shows epHAL model significantly outperformed both pHAL and conventional HAL. Figure 1 summarizes the trends in which the Information Retrieval Modeling techniques are enhancing/upgrading their capabilities by covering more and more semantic information and adopting better representation scheme.

4. CONCLUSION
In this paper various information indexing and retrieval techniques (based on both statistical methods and language processing techniques/approaches) are, first, discussed briefly and then a comparative study of these is presented. It helped us to identify the strength and weakness of various techniques and the research trends shifting in the domain of web information effectively. The study suggests that the retrieval systems can be more efficient if we use more and more semantic knowledge and Natural Language Processing techniques. This paper may serve the purpose of ready references for the naive researchers.

REFERENCES


Vector space modeling techniques (VSM)

Hyperspace analog to Language Modeling (HAL)

Language modeling techniques (LM)

Best-Match Retrieval Techniques

Exact-Match retrieval Techniques

Figure 1. Trends in IRM Techniques