Ranking Strategy Using Hybrid Model

Om Vikas
Senior Member, IEEE
Advisor to C-DAC, India

Pooja Arora
Assistant Professor
AKGEC Ghaziabad, India

ABSTRACT
Various information retrieval models generate different ranking list as output. This paper presents the comparative analysis of the vector space model and the probabilistic model. Effect of stopword removal is also discussed. A new hybrid model is introduced that combines the Vector Space Model and the Probabilistic model. The resultant model gives better performance. For experiments, we have constructed English-Hindi IR test collection from EMILLE parallel corpus. Relational (stop) words are considered for improving the search results. F-measure and AIP (Average Interpolated Precision) are used for evaluation.

General Terms
Information Retrieval, Non-relational Stopwords, IR system architecture

Keywords
IR models comparison, Stopword removal, English-Hindi parallel corpus, Relational Stopwords, Hybrid Model, Vector Space Model, and Probabilistic Model.

1. INTRODUCTION
Information Retrieval (IR) is the method of searching documents, and information within documents and metadata about documents in databases and on the World Wide Web. IR deals with the representation, storage, organization of, and access to information items. The representation and organization of the information items should provide the user with easy access to the information in which he is interested. It is defined as the finding documents of an unstructured nature (usually text) that satisfies information need from within large collections.

Information Retrieval is different from data retrieval. Data retrieval mainly consists of determining which documents of a collection contain the keywords in the user query which is not enough to satisfy the user information need. In fact, the user of an IR system would be concerned more with retrieving information about a subject rather than retrieving data which satisfies a given query. For an Information Retrieval system, the retrieved object might be inaccurate and small errors are likely to go unnoticed but for a data retrieval system, however a single erroneous object among a thousand retrieved objects means total failure. Data retrieval is concerned more with syntactic precision where as information retrieval is concerned more with semantic relevance.

2. IR MODELS
Although a variety of models have been developed to retrieve information, Vector Space Model (VSM) and Probabilistic Model are the two leading models in IR systems.

In Vector Space Model, documents and queries are represented as vectors in a common vector space. Retrieval is based on the cosine similarity between the query vector and the document vector that could be used as a measure of the score of the document for that query.

In Probabilistic Model, retrieval is based on whether a probability of relevance of a document is higher than that of non-relevance. The document with probabilities of relevance at least that of irrelevance are ranked in decreasing order of their relevance.

Each model has its own advantages. This paper studies the effect of both the models on the IR system and also the possibility of combining the models to retrieve more relevance documents in the top of the ranked list. A hybrid model of Vector Space Model and Probabilistic Model is suggested.

English
Document name: eng-w-social-financial
Query: *Security at workplace*
Hindi
Document name: hin-w-social-financial
Query: *आमदनी के लिए सहायता*

Let,
Term_Frequency(tf): No. of times term t present in the document
Maximum_Term_Frequency(max_tf): Maximum term frequency of the term t among all the documents
Document_Collection(N): No. of documents in the collection = 46
Relevant_Document(S): No. of document relevant to the query = 5 (for both the queries)
Relevant_Document_Contain_Term(s): No. of relevant document containing term t
Document_Frequency(n): No. of documents containing term t
Maximum_Score_Vector(max_score$_v$): Maximum score of the document retrieved using VSM
Maximum_Score_Probabilistic(max_score$_p$): Maximum score of the document retrieved using probabilistic model

Table 1 Different Parameters for English Document

Terms	tf	max_tf	s	n
Security (t1)	10	12	4	14
at workplace (t2)	6	8	2	10

Table 2 Different Parameters for Hindi Document

Terms	tf	max_tf	s	n
आमदनी के लिए (t1)	8	12	4	12
सहायता (t2)	9	14	2	10

2.1 Vector Space Model

The representation of a set of documents as vectors in a common vector space is known as the Vector Space Model. The vector $\vec{V}(d)$ is derived from the document d with one component in the vector for each dictionary term i.e. the size of the vector is equal to the size of the dictionary. The component may be computed using tf weighting scheme. We have computed weight of the i^{th} term in the j^{th} document as:

$$w_{ij} = \frac{tf_{ij}}{max\_tf_i} \quad (1)$$

where tf_{ij} is the Term_Frequency of i^{th} term in j^{th} document, max_tf_i is the Maximum_Term_Frequency of the i^{th} term.

The set of documents in a collection may be viewed as a set of vectors in a vector space, in which there is one axis for each term. A query can also be viewed as a vector of very short document. We have computed weight of the i^{th} term in the query as:

$$wq_i = \frac{1}{Term\_Occurence} \quad (2)$$

where Term_Occurence is the number of times term is present in the query

The cosine similarity function between the query vector & the document vector is used as a measure of the score of the document for that query.

Thus, $sim(d, q) = \vec{d} \bullet \vec{q}$ (3)

where $\vec{d}$, $\vec{q}$ are the two unit vectors representing query and document respectively.

The document score is calculated as:
$Score_v = sim(d, q)$
where, $Score_v$ is the similarity score of the document using VSM
The resulting score can thus be used to select the top-scoring document for a query.

To illustrate above mentioned document and query in:

English

$\vec{d}$ = <$\frac{10}{12}$, $\frac{6}{8}$> = <0.83, 0.75> $\vec{q}$ = <1, 1>

Therefore, sim(d ,q) =0.83+0.75 = 1.58
Thus, Score = 1.58
Document vector for English Document that contains both terms with maximum frequency = <$\frac{12}{12}$, $\frac{8}{8}$> = <1, 1>

Therefore, maximum score of the document (max_score_v) = 1+1 =2

Normalized score = $\frac{Score}{Maximum\_score\_vector} = \frac{1.58}{2} = 0.79$

Hindi

$\vec{d}$ = <$\frac{8}{12}$, $\frac{9}{14}$> = <0.67,0.64> $\vec{q}$ = <1,1>

Therefore, sim(d ,q) =0.67+0.64 = 1.31
Thus, Score = 1.31
Document vector for Hindi Document that contains both terms with maximum frequency = <$\frac{12}{12}$, $\frac{14}{14}$> = <1, 1>

Therefore, maximum score of the document (max_score_v) = 1+1 =2

Normalized score = $\frac{Score}{Maximum\_score\_vector} = \frac{1.31}{2} = 0.65$

2.2 Probabilistic Model

IR system deals with uncertain information, so probability theory seems to be the most likely way to enumerate uncertainty. In this model, retrieval is based on whether a probability of relevance of a document is higher than that of non-relevance. Ratio of the probability of relevance to the probability of non-relevance of the document is used to measure the similarity between document and query. Similarity function may be defined as:

$$sim_p(d, q) = \frac{Prob\_of\_relevance}{Prob\_of\_nonrelevance} = \sum_{t \in Q} weight(t) \quad (4)$$

where, $$weight(t) = \log \frac{(s)/(S-s)}{(n-s)/[(N-n)-(S-s)]} \quad (5)$$

s/(S-s) = ratio that the relevant document contain the term
(n-s)/[(N-n)-(S-s)] = ratio that the non-relevant document contains the term
Let,
Term (t) is the term present in the query Q,
Document_Collection(N) is the number of documents in the collection,
Document_Contain_Term(n) is the number of documents containing the term t,
Relevant_Document(S) is the number of documents relevant to the query,
Relevant_Document_Contain_Term(s) is the number of relevant documents containing the term t.
The document score is calculated as:

$$Score_p = \sum_{t_i \in Q} weight(t_i) \quad (6)$$

where, $weight(t_i)$ is weight of the term as calculated by the equation (5).

$Score_p$ is the score of the document using probabilistic model
If the term is not present in the document then the term weight is zero for that particular term.
To illustrate above mentioned document and query in:
English

$$weight(t1) = \log \frac{(4)/(5-4)}{(14-4)/[(46-14)-(5-4)]} = 7.10$$

$$\text{weight}(t\ 2) = \log \frac{(2)/(5-2)}{(10-2)/[(46-10)-(5-2)]} = 2.75$$

Thus, Score = 7.10+2.75 = 9.85
Therefore, maximum score of the English document using probabilistic model is also 9.85 which is the score of the document containing both the terms. The score of the document containing term t1 and not term t2 is 7.10 and document containing term t2 and not containing term t1 is 2.75.
Thus, max_score_p for English document = 9.85

Hindi

$$\text{weight}(t\ 1) = \log \frac{(4)/(5-4)}{(12-4)/[(46-12)-(5-4)]} = 16.5$$

$$\text{weight}(t\ 4) = \log \frac{(2)/(5-2)}{(10-2)/[(46-10)-(5-2)]} = 2.75$$

Thus, Score = 16.5+2.75 = 19.25
Therefore, maximum score of the Hindi document using probabilistic model is also 19.25 which is the score of the document containing both the terms. The score of the document containing term t1 and not term t2 is 16.5 and document containing term t2 and not containing term t1 is 2.75.
Thus, max_score_p for Hindi document = 9.85

Thus, Normalized score of the document can be calculated as:

$$\frac{\text{Score}}{\text{Maximum\_score\_Probabilistic}}$$

3. IMPLEMENTING IR SYSTEM

We have implemented information retrieval system in Java. The various components of the Information Retrieval system are as follows:

3.1 Pre-processing Steps

The various pre-processing steps are performed on the documents:

3.1.1 Markup and Format Tags Removal

During this phase, all markup tags and special formatting are removed from the document. Thus, for an HTML document all tags and text inside these are removed. This normally would include all element attributes, scripts, comment lines and text placed into these.

3.1.2 Tokenization

During this phase, all remaining text is broken up into pieces called tokens and at the same time it throw away certain characters such as punctuation marks. These tokens are often loosely referred to as terms or words.

3.1.3 Stopword Removal

Stopwords are the common words that appear in the text. There are two types of stopwords – Relational (in, on, under, near, at) & Non-relational (are, the, am, is, a, an). These stopwords have different impact on the information retrieval process. Relational stopwords indicate semantic relevance that is necessary for efficient IR. Removing relational stopwords from the document would result in loss of such relevant semantic information resulting in decrease of relevance efficiency of the system. Removing non-relational stopwords would reduce the document length resulting into faster search.

For example: **In English -** in, on, upon (relational) are, the, am, a, an (non-relational)

In Hindi – का, में, से, नीचे (relational) है, था, रहा, सकता (non-relational)

In our approach, we remove only non-relational stopwords. We apply term merging technique to merge the relational terms with the previous term in the text and then perform indexing accordingly. This step would help in maintaining the relationship between the terms.

For example: In *आमदनी के लिए सहायता*, we merge *के लिए* with *आमदनी* to make a single combined term *आमदनी के लिए.*

In *security at workplace***,** we merge *at* with *workplace* to make a single combined term *at workplace*.
The stopwords that are used to mark the various semantic roles are categorized as relational stopwords.
For example: "to", "into", "toward" mark the goal or end point of movement or transition, "from", "by" mark the source or starting point of movement or transition, "near", "in", "at" mark the physical location where an action occurs or that action is related to, " using", "with" mark the instrument or tool or intermediary to be used etc.

3.1.4 Stemming

It refers to the process of reducing terms to their stems or root variant. Thus, " किताब ", " किताबें "and " किताबों " are reduced to " किताब ""(**in Hindi**) and "concatenate", "concatenated" and "concatenates" are reduced to "concatenate" (**in English**). This step helps in increasing the recall rate. After stemming, the key terms of a query or documents are represented by stems rather than by the original terms.

3.2 Indexing

It is the way documents are managed in the collection. To make searching more efficient, a retrieval system stores documents in an abstract representation. The set of keywords are stored, along with links to the document in which each word appears. This structure for storing indexing information is called an index file. For each word in the document one entry is stored in the index file. Each entry contains list of the documents in which that word is present along with its positions in that particular document. Format of the entry in the index file:
w1 {d1 wt1 freq1 : < pos11,pos12,…..>, d2 wt2 freq2 : <pos21,pos22,……> . . . . . .so on}
where d1, d2… are the documents which contains word w1
wt1, wt2… are the weights of w1 in the respective document
freq1, freq2… are the frequencies of w1 in the respective document
pos11, pos12.. are the positions of w1 in the first document
pos21, pos22… are positions of w1 in the second document and so on.
Index files were prepared for sample documents in English and Hindi.

3.3 Searching

It retrieves documents that contain a given query token from the inverted index.

For the above mentioned query, it will extract entry for **income** and **support** from the index file. (**English**)

income

doc1 0.40 5: <45,57,83,98,163>
doc2 0.10 2: <61,87>
doc3 0.10 2: <14,43>
doc4 0.10 2: <7,51>
doc5 0.14 3: <34,65,81>
doc6 0.83 8: <20,45,64,104,135,172,188,203>

support

doc1 0.20 3: <21,45,89>
doc2 0.20 2: <10,54>
doc3 0.60 6: <15,34,63,99,142,199>
doc4 0.10 1: <24>
doc5 0.30 3: <27,52,73>
doc6 0.75 7: <21,32,65,136,160,189,191>

It will extract entry for आमदनी के लिए and सहायता from the index file. (**Hindi**)

आमदनी के लिए

doc1 0..67 5:<10,45,76,112,123>
doc2 0.0010 1:<25>
doc3 0.0010 1:<43>

सहायता

doc1 0.64 4:<7,15,45,61>
doc2 0.0010 1:<16>
doc3 0.0020 2:<16,32>
doc4 0.01 3:<13,42,65>
doc5 0.30 3:<34,54,7

3.4 Ranking

It scores all retrieved documents according to the weighting schemes mentioned above. For each document present in the extracted entries, it will calculate Vector Space Model score and Probabilistic Model score and then combine the two scores using suggested hybrid to get the final score of the document. And finally, the retrieved documents are sorted based on the calculated score.

Initial Document Set: { doc1, doc2, ..., docN}
Retrieved Documents: { doc1, doc5, doc8, doc15}
Sorted Documents: { doc8, doc1, doc5, doc15 }

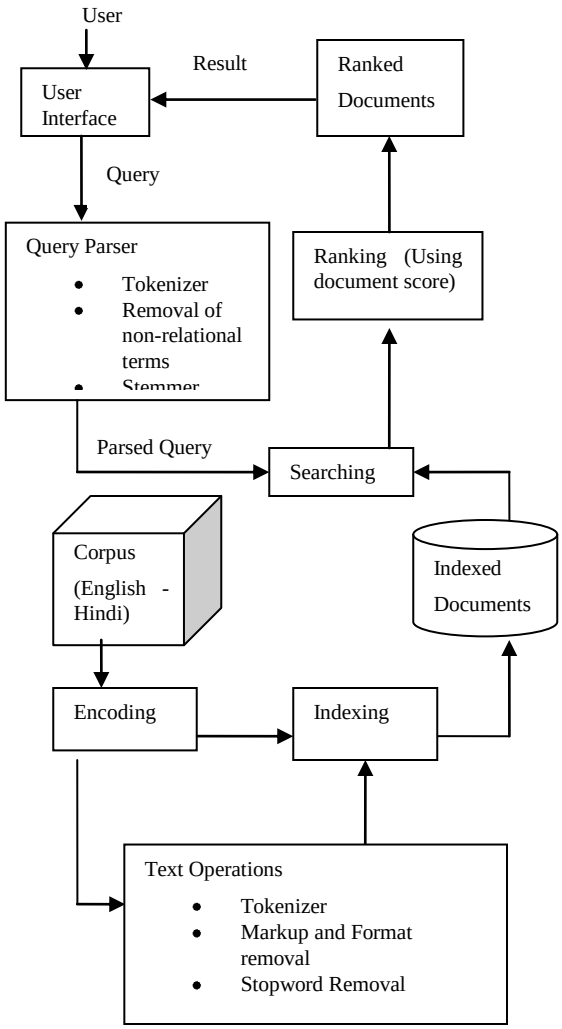

Figure 1 The Architecture of IR system

3.5 User Interface

Interface manages interaction with the user. It takes query as input from the user and display ranked list of documents as output.

4. EVALUATION MEASURES

4.1 Recall (R)

It is the measure of the ability of a system to present all relevant items.

$$R = \frac{\text{Number of relevant documents retrieved}}{\text{Number of relevant documents in collection}}$$

4.2 Precision (P)

It is the measure of the ability of the system to present only relevant items.

$$P = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of documents retrieved}}$$

To evaluate ranked lists, precision can be plotted against recall after each retrieved document. To facilitate computing average performance over set of queries – each with different number of relevant documents – precision values for individual query are interpolated to a set of standard recall levels (0 to 1 in increments of .1). The standard rule to interpolate precision at standard recall level I is to use the maximum precision obtained for the query for any actual recall level greater than or equal to i.

Mathematically, Interpolated precision $P_{interpolated}$ at certain standard recall level i is defined as the highest precision found for any recall level $i' \geq i$:

$$P_{interpolated(i)} = \max P(i') \quad i' \geq i$$

4.3 Average Interpolated Precision (AIP)

It is the average of the interpolated precision at each standard recall point value for all queries together.

$$AIP_i(Q) = \frac{1}{|Q|}\sum_{j=1}^{|Q|} P_{interpolated(i)}(j) \qquad 0.0, 0.1, 0.2, \ldots, 1.0$$

where Q is the set of queries,
$P_{interpolated(i)}$ is the interpolated precision at i^{th} recall value level,
$AIP_i(Q)$ is the average of the interpolated precision at i^{th} recall level for all the queries in set Q.

4.4 F-Measure (F)

It is the harmonic mean of precision and recall. It is a single measure that trades of precision versus recall.

$$F = \frac{2*\text{Precision}*\text{Recall}}{(\text{Precision} + \text{Recall})}$$

4.5 Mean Average Interpolated Precision (MAIP)

It is the mean of all the average interpolated precisions calculated at all standard recall points.

$$MAIP = \frac{\sum AIP_i(Q)}{No\_recall\_points} \qquad i = 0.0, 0.1, 0.2, \ldots, 1.0$$

where $AIP_i(Q)$ is the average of the interpolated precision at i^{th} recall level for all the queries in set Q
No_recall_points are the number of standard recall points used.

5. CORPUS AND EVALUATION

For experiments, we have created English-Hindi test collection of about 46 documents extracted from EMILLE corpus and also generated 20 queries along with their relevant documents for testing. All data are encoded in Unicode text.

Table 3 Test Collection Detail

	English	Hindi
No. of Documents	46	46
No. of index terms	7216	11889
No. of queries	20	20
Average No. of terms/doc	186	258

6. HYBRID MODEL

It is clear from the graph that probabilistic model perform better for lower recall values, which bring to higher precision while Vector Space Model perform better for higher recalls. We can combine the two models by combining the score of the document for the two models to create a new rank list.
Let, s_{vnorm} = normalized score of a document in Vector Space Model
s_{pnorm} = normalized score of a document in Probabilistic model

$$score_{new} = (s_{vnorm} + s_{pnorm}) / 2 \qquad (7)$$

where, $score_{new}$ is the combined score of the documents which is used to rank the documents.

MAIP of the three models:

Vector Space Model 0.47
Probabilistic Model 0.54
Hybrid Model 0.59

F-measure values and average interpolated precision values at various recall points are also improved using Hybrid Model (as shown in the graph)
In the Hybrid model, the average interpolated precision values are further improved by using our term merging approach.(as shown in the graph).

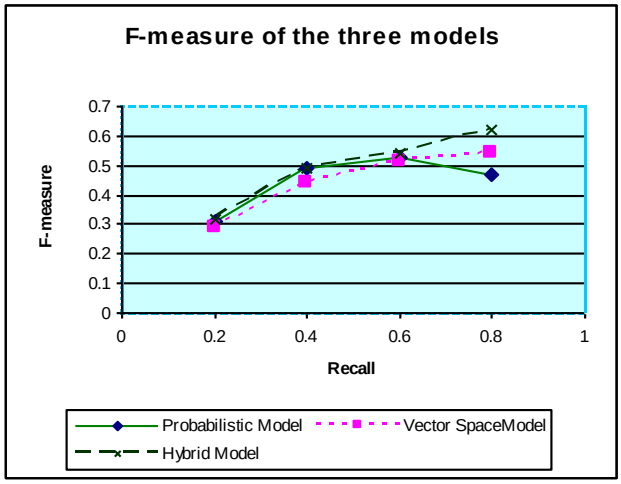

Figure 2 F-measure of the three models

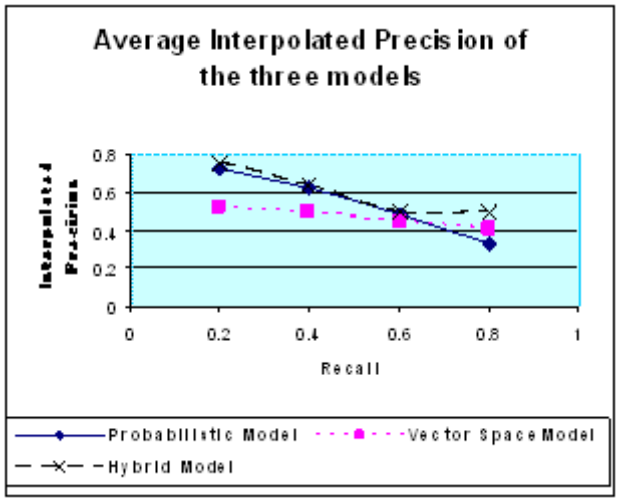

Figure 3 Average interpolated precision of the three models

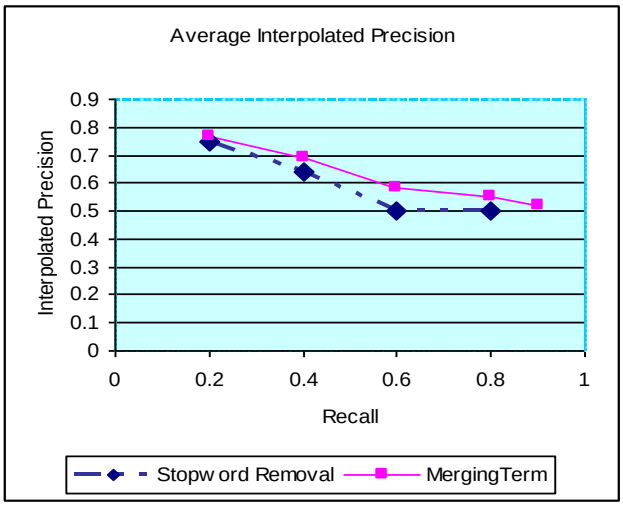

Figure 4 Average interpolated precision comparing our merging technique with conventional stopword removal

7. CONCLUSION

In this paper we have performed comparative analysis of Vector Space model and Probabilistic model. Results show that there is a need for a combined model that comprises features of both the models. We suggest a new hybrid model that combines a Vector Space Model and a Probabilistic model. Performance of the hybrid model is better than both the models. Effect of stopword removal is also discussed. Removal of relational stopwords would simply results in loss of relation that exists between terms in the documents. We have discussed a new technique which removes non-relational stopwords only and merges the relational stopword with preceding/following term as per language grammar to maintain the existing relation and then perform indexing of the document. This technique has great effect on the performance of the system. For experiments, we have constructed English-Hindi IR test collection from EMILLE parallel corpus. F-measure and AIP (Average Interpolated Precision) are used for evaluation.

8. REFERENCES

[1] G. Salton, A. Wong and S.S. Yang 1975. A vector space model for automatic indexing, communications of the ACM, 18, pages 613-620.

[2] L. Gravano, H. Garcia-Molina.1997. Merging Ranks from Heterogeneous Internet Sources. Very Large Databases (VLDB).

[3] Ashwani Mujoo, Manoj Kumar Malviya, Rajat Moona, T.V. Prabhakar.2000. A Search Engine for Indian Languages, In Proceedings of the First International Conference on Electronic Commerce and Web Technologies, pages 349-358.

[4] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze.2008.Introduction to Information Retrieval, Cambridge University Press.

[5] S.E. Robertson, K. Sparck Jones.1976. Relevance Weighting of Search Terms, Journal of the American Society for Information Science, pages 129-146.

[6] The EMILLE Corpus, http://ahds.ac.uk/catalogue/collection.htm?uri=lll-2460-1

[7] Prasad Pingali, Jagadeesh Jagalamudi, Vasudeva Varma.2006. Webkhoj : Indian Language IR from multiple character encodings, In Proceedings of the 15th international conference on World Wide Web , ACM , pp : 801-809.

[8] Amaresh Kumar Pandey , Tanveer J Siddiqui.2008. An Unsupervised Hindi Stemmer with heuristic improvements, In the proceedings of the 2nd workshop on Analytics for noisy unstructured text data, ACM , pp : 99-105.

[9] M.F. Porter.1997. An Algorithm for Suffix Stripping.1997. Morgan Kaufmann Multimedia Information And Systems Series , pp :313-316.