

Relation Inclusive Search for Hindi Documents

Pooja Arora

Research Scholar, Banasthali Vidyapith
Assistant Professor, MCA Department AKGEC
Ghaziabad, India

Om Vikas

Former Director, ABV, IITM, Gwalior
Senior Member, IEEE
India

Abstract—Information retrieval (IR) techniques become a challenge to researchers due to huge growth of digital and information retrieval. As a wide variety of Hindi Data and Literature is now available on web, we have developed information retrieval system for Hindi documents. This paper presents a new searching technique that has promising results in terms of F-measure. Historically, there have been two major approaches to IR - keyword based search and concept based search. We have introduced new relation inclusive search which performs searching of documents using case role relation, spatial relation and temporal relation of query terms and gives results better than previously used approaches. In this method we have used new indexing technique which stores information about relation between terms along with its position. We have compared four types of searching: Keyword Based search without Relation Inclusive, Keyword Based search with Relation Inclusive, Concept Based search without Relation Inclusive and Concept Based search with Relation Inclusive. Our proposed searching method gave significant improvement in terms of F-measure. For experiments we have used Hindi document corpus, Gyannidhi from C-DAC. This technique effectively improves search performance for documents in English as well.

Keywords—Relation inclusive search; RSearch; spatial & temporal prepositions and postpositions; Hindi document retrieval; case roles.

I. INTRODUCTION

World Wide Web is used to share huge repository of texts in many languages as everybody can post content in any language. Search across multiple languages is desirable with the increase of many languages on the web. In order to enable a wider proportion of population to benefit from Information technology, it is desirable that human-machine interface permits one's native language of communication. In the context of a multi-lingual country like India, this can be of immense value. The users will be greatly helped if they are able to retrieve information in a language in which they are comfortable. In the context of Indian languages, Hindi language has been given much emphasis leading to the development of significant number of Hindi documents. According to Ethnologue statistics, among the list of the top 10 languages, Hindi is coming forth with 260 million first language speakers and English is coming third with 335 million first language speakers [14]. Hindi language is spoken by 41% of population of India whereas about 5% of population understands English as their second language [10]. A wide variety of Hindi Data and Literature is now available on web [6]. The number of users who want the information in Hindi language is increasing. We have developed an Information Retrieval System (IRS) for Hindi documents. Various search engines are available on the internet as independent search

engine sites in Hindi like Google [11], Raftaar [12] and Hinkhoj [13]. The retrieval accuracy of these search engines does not satisfy users need. Increasing the appropriateness of the results returned by these search engines is critical to dealing with the huge repository of the data.

Historically, there have been two major approaches to IR – **keyword based search** and **concept based search**. In keyword based search, search engines use words or multi-words phrases that occur in documents and queries as atomic elements and the content of the documents are described by list of keywords. The search procedure, used by these search engines, is principally based on the exact matching of document and query terms and does not take into consideration the various meanings or possible concepts that a word represents [4], [5]. If the user chooses a valid synonymous word that is not in any document then it would fail the search. In general, this approach has many problems as the user may not get the most relevant and useful content related to the query. The first problem is low precision, which is due to the irrelevance of many of the search results. This results in a difficulty finding the relevant information. The second problem is low recall, which is due to the inability to index all the information available on the Web. This results in a difficulty finding the unindexed information that is relevant. Also the level of deepness of analysis of the language is very low, so the relevant information is not retrieved by the search engines.

On the other hand, concept based search fetch document based on their meaning rather than the presence of the keywords. Here, the meaning of words is analysed and not only their syntactic representations. This type of searching uses query expansion techniques where the initial query is appended with related, contextual, or synonymous terms so as to make the new query more complete to define the required concept [6], [7], [8]. But, this approach also uses the words that occur in query and in documents as atomic elements and no relation between them is used for retrieval. However, concept-based approaches allow reaching a higher precision than keyword based approaches.

There is a need to consider relation between terms present in the documents. Documents could contain three types of relations: **case role relation**, **spatial relation** or **temporal relation**. For example, “*टेबल के ऊपर किताब*” (*table ke upar kitaab*) => “*book on table*” contains spatial relation i.e. position of *किताब* w.r.t *टेबल* given by *के ऊपर* (postposition) and position of *book* w.r.t. *table* given by *on* (preposition); “*श्याम ने राम को किताब दी*” (*Shyam ne Ram ko kitaab di*) contains case role (karaka) relation. Here, case role of *श्याम* (*Shyam*) is *कर्ता* (nominative case) and *राम* (*Ram*) is *कर्म* (objective case); “*राम*

से पहले श्याम घर जाएगा” (Ram se pehle Shyam ghar jayega) contains temporal relation given by “पहले”. If the documents are indexed without considering relational information and remove prepositions/postpositions as stopwords before indexing of the documents to reduce index file size, then the relation existing between terms is lost. Taking this point into account, we propose **relation inclusive search** as a new and promising way of improving search on the IRS. We call it **RSearch** (relation inclusive search). The main idea is to keep the same infrastructure which has made previous methods so successful, thus improving the system performance. Informally, **RSearch** do not use terms present in query and in documents as atomic elements, but the semantics of relationships between query terms is considered for improving search. While indexing it stores all the relational information existing between terms present in the documents in the index file and use that information for retrieving relevant documents.

To reduce the size of index file we have removed stop words before indexing the documents. We have categorized stopwords in two categories – Relational (नीचे, ऊपर, पर, आगे, अंदर, ने, को, है, था etc) & Non-relational (ही, तब, यह etc). These stopwords have different impact on the information retrieval process. Relational stopwords indicate semantic relevance that is necessary for efficient information retrieval. Removing relational stopwords from the document would result in loss of such relevant semantic information resulting in decrease of relevance efficiency of the system. While removing non-relational stopwords would reduce the document length resulting into faster search. So, we remove only non-relational stopwords.

II. RELATION INCLUSIVE SEARCH

There are multiple objects in a query which are dependent on each other. Relations between objects are given by prepositions (in English) and postpositions (in Hindi) depending upon the language of the document. We have proposed a new relation inclusive searching technique **RSearch** which stores relation between terms present in the document in the index file and also considers that relation for retrieving relevant documents from the corpus. It is based on new indexing scheme. It fetches documents based on **Case Role (Karaka) relation, Spatial relation** and **Temporal relation** existing between query terms. All these types of relations are discussed in the following subsections in detail. Before that, let us consider the document collection shown in figure 1. In figure 2, we showed examples of ten queries, which are submitted to this document collection. We will use the same document collection and sample queries in the following subsections.

III. TYPES OF RELATIONS

There are three types of relations we have considered in relation inclusive search.

A. Case Role (Karaka) relation

Hindi language has eight types of case roles which are shown below with related परसर्ग/suffixes. (see table I) [9]

D1: दो दिन पहले बारिश हुई थी do din pehle barish huee thee
D2: किताब टेबल पर रखी है kitaab table par rakhi hai
D3: हिंदी की किताब टेबल पर रखी थी Hindi kee kitaab table par rakhi thee
D4: एक कुत्ता टेबल के नीचे से कूदकर भाग गया ek kutta table ke niche se koodkar bhaag gaya
D5: श्याम ने एक जानवर को मारा Shyam ne ek janwar ko mara
D6: श्याम के बाद राम घर जाएगा Shyam ke baad Ram ghar jayega

Fig.1. A document collection

Query #	Query	Type of relation present
1	दो दिन के बाद बारिश होगी do din ke baad barish hogi	Temporal relation
2	वृक्ष के पास जानवर vriksh ke paas janwar	Spatial relation
3	हिन्दी की पुस्तक टेबल के नीचे है Hindi kee pustak table ke niche hai	Spatial relation
4	टेबल पर किताब रखी है table par kitaab rakhi hai	Spatial relation
5	राम निवास के पास खड़ा है Ram niwaas ke paas khara hai	Spatial relation
6	जानवर ने श्याम को मारा janwar ne Shyam ko mara	Case role relation
7	औरत ने चाकू से फल काटा aurat ne chaku se phal kata	Case role relation
8	डिब्बे के अंदर पुस्तक dibbe ke ander pustak	Spatial relation
9	रूम में टी. वी. रखा है room mein T.V. rakha hai	Spatial relation
10	राम से पहले श्याम घर जाएगा Ram se pehle Shyam ghar jayega	Temporal relation

Fig.2. Sample queries

Consider the following sentence:

Sentence 1: जानवर ने श्याम को मारा ।

janwar ne Shyam ko mara
Animal killed Shyam.

Here, case role of जानवर is कर्ता (nominative case) and श्याम is कर्म (objective case).

As Hindi is free order language; order of words contains only secondary information such as emphasis etc. Primary information relating to ‘gross’ meaning (e.g., one that includes semantic relationships) is contained elsewhere [1], [2], [3].

Therefore, same sentence could be written as:

Sentence 2: श्याम को जानवर ने मारा ।

An answer for query Q6 in figure 2 using keyword based searching would be D5 as this document contains all the query terms. But this is not the correct answer as case roles of जानवर and श्याम do not match. In D5, the case role of श्याम is कर्ता (nominative case) and जानवर is कर्म (objective case) but reverse is the case in Q6.

So, to improve the information retrieval effectiveness in terms of precision and recall, we have stored the case role relation between terms in the index file.

1) Indexing a Case Role relation

While indexing the documents, the **case role** (giving the karaka relation) of the word is also added in its posting list along with its position in a particular document. For example,

TABLE I. CASE ROLES WITH RELATED SUFFIXES

S.No.	कारक / Case Role	परसर्ग (In Hindi)	Suffixes (In English)
(i)	कर्ता (karta)/ nominative case (K1)	ने, ne	by
(ii)	कर्म (karma)/ objective case (K2)	को, ko	to
(iii)	करण (karana)/ instrumental case (K3)	से (se), के साथ (ke sath), के द्वारा (ke dvara)	with
(iv)	संप्रदान (sampradaan)/ dative case (K4)	के लिए (ke liye), को (ko)	for, to
(v)	अपादान (apaadaan)/ ablative case (K5)	से (se)	taking from (separation / detachment)
(vi)	सम्बन्ध (sambandh)/ relative case (K6)	का (ka), के (ke), की (ki), रा (ra), रे (re), री (ri)	belong to (something / somebody)
(vii)	अधिकरण (adhikarana)/ locative case (K7)	में (mein), पर (par)	in, on
(viii)	सम्बोधन (sambodhan)/ vocative case (K8)	हे! (hey!), अरे! (are!)	oh!

जानवर is present at 4th position in document D5 with case role as **कर्म (K2)** is represented as D5:[[4,K2]] and **श्याम** is present at 1st position in document D5 with case role as **कर्ता (K1)** is represented as D5:[[1,K1]]. (see figure 3)

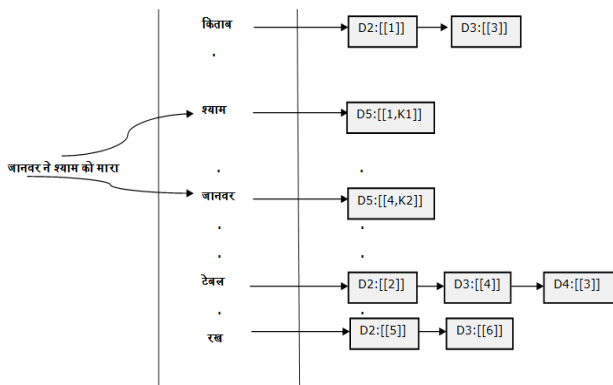


Fig.3. Inverted Index file in relation inclusive search

2) Steps for matching

To retrieve the documents for query Q6, matching involves the following steps:

- 1) Retrieve the posting lists of **जानवर** and **श्याम**.
- 2) Get the set of documents from the posting lists containing both the terms and also their case roles in those documents.
- 3) Find out the case roles of query terms.

4) For each document check if the case roles of both the terms are matching with their case roles present in the query.

In this example, document D5 contains both the terms. In D5, case role of **जानवर** is **कर्म (K2)** and that of **श्याम** is **कर्ता (K1)**. But the **जानवर** is **कर्ता (K1)** and **श्याम** is **कर्म (K2)** in Q6. As the case roles of **जानवर** and **श्याम** are not matching, an answer for query Q6, computed by relation based search is the empty set which is the correct answer.

B. Spatial relation

Spatial relation shows position of an object w.r.t another object.

For example,

Document D2 gives position of **किताब** w.r.t **टेबल** (see figure 1).

1) Indexing a Spatial relation

All the relational stopwords giving spatial relation are indexed in separate inverted file. For example, in posting list of **पर** - D2:[[2,1]] means in document D2 **पर** gives the relation between words present at positions 2 (**टेबल**) and 1 (**किताब**) respectively (see figure 4).

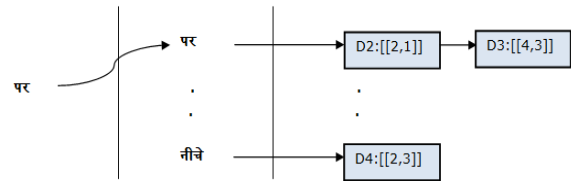


Fig.4. Inverted Index of Relational Stopwords

2) Representation of Spatial relation

We are extracting relations in the form of triples:

Word₁=>RELATION<=Word₂ and they are used as indexing terms.

For example:

In documents D2 and D3, relation **टेबल =>पर<= किताब** exists, that means **किताब** is lying on the **टेबल**.

In document D4, **टेबल =>नीचे<= कुत्ता** exists, that means **कुत्ता** is sitting below the **टेबल**.

3) Spatial relations extracted

Currently, we have extracted the following relations:

- 1) नीचे (below)
- 2) ऊपर/पर (on)
- 3) पास (near)
- 4) दूर (far)
- 5) आगे (front)
- 6) पीछे (back)
- 7) अंदर (inside)
- 8) बाहर (outside)

4) Properties of Spatial relation

Following are the properties of spatial relation:

1. If relation $X \Rightarrow REL1 \Leftarrow Y$ exists, then relation $Y \Rightarrow REL2 \Leftarrow X$ also exists such that REL1 is antonym (opposite relation) of REL2.

For example, in document D2 if relation टेबल \Rightarrow पर \Leftarrow किताब exists, then relation किताब \Rightarrow नीचे \Leftarrow टेबल also exists as relations पर and नीचे are opposite relations.

Various opposite relation are:

- ऊपर/पर opposite of नीचे
- पास opposite of दूर
- आगे opposite of पीछे
- अंदर opposite of बाहर

2. If relations $X \Rightarrow REL \Leftarrow Y$ and $Y \Rightarrow REL \Leftarrow Z$ exists then relation $X \Rightarrow REL \Leftarrow Z$ also exists.

For example, if relations टेबल \Rightarrow पर \Leftarrow किताब and किताब \Rightarrow पर \Leftarrow कलम exists, then relation टेबल \Rightarrow पर \Leftarrow कलम also exists.

3. If relation $X \Rightarrow REL \Leftarrow Y$ exists, then relation $Y \Rightarrow REL \Leftarrow X$ also exists for relations in set $REL = \{\text{पास, दूर}\}$

For example, if relation राम \Rightarrow पास \Leftarrow रहीम exists, then relation रहीम \Rightarrow पास \Leftarrow राम also exists.

5) Steps for matching

For example,

Consider the query Q4 in figure 2. Following steps are involved for searching of relevant documents:

Steps:

1) Perform stemming on query terms. Query Q4 reduced to: टेबल पर किताब रख (see figure 5).

2) Retrieve posting lists of query terms.

3) Using posting lists, we get the document set containing query terms टेबल, पर, किताब and रख {D2, D3}. We left D4 as it does not contain terms किताब, पर and रख.

4) Match query relation with the relation present in the resultant documents. Relation in query is टेबल \Rightarrow पर \Leftarrow किताब.

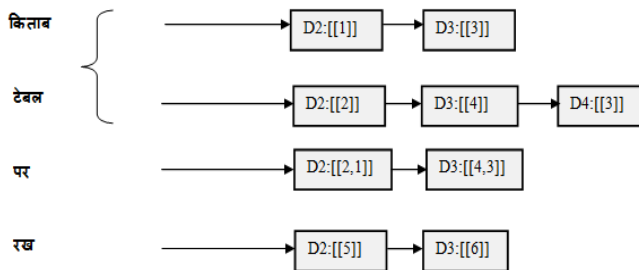


FIG.5. Posting lists of query terms

Now the relation retrieved from posting lists is checked. Posting list of पर contains D2:[2,1], D3:[4,3]. That means D2 contains relation पर between words at position 2 and 1 which are टेबल and किताब respectively giving relation टेबल \Rightarrow पर \Leftarrow किताब. Similarly, D3 contains relation पर between words at position 4 and 3 which are टेबल and किताब respectively giving relation टेबल \Rightarrow पर \Leftarrow किताब. As the relation is matching in both the documents, {D2, D3} resultant document set is retrieved.

C. Temporal relation

All the temporal related words (पहले, बाद मे) come under relational stop words.

For example,

दो दिन पहले बारिश हुई थी।

do din pehle barish huee thee

Here, “दो दिन पहले” gives the temporal relation. If we remove it, the temporal relation would be lost and the system does not retrieve relevant document. In order to improve the efficiency of the system we don’t discard temporal related words (पहले, बाद मे) as stop words.

1) Indexing a Temporal relation

All the relational stopwords giving temporal relation are indexed in separate inverted file. The posting lists of temporal related words contain temporal information. We have considered two types of temporal relation- **Number_of_days** & **Order_of_entities**.

1. Number_of_days: If the document/query contains words like {दिन, दिवस} along with words {पहले, बाद}, then the temporal relation is “**Number_of_days**”. In that case we store number of days in the posting list of temporal related words along with their positions.

For example,

In document D1, temporal related word पहले is present at 3rd position and number of days are 2. Same information is stored in the posting list of पहल (stemmed word) (see figure 6).

2. Order_of_entities: If the document/query contains words like से, के along with words {पहले, बाद}, then the temporal relation is “**Order_of_entities**”. In that case we store order of entities in the posting list of temporal related words along with their positions. However, if the query contains temporal related word पहले, then we check for the posting list of बाद also and vice versa as पहले and बाद are opposite words.

For example,

In document D6, temporal related word बाद is present at 3rd position giving the order of entities between श्याम & राम which are stored at 1st and 4th position respectively. Same information is stored in the posting list of बाद. Relation between श्याम & राम is श्याम \Rightarrow बाद \Leftarrow राम (see figure 6).

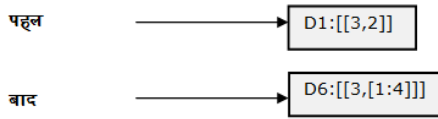


Fig.6. Indexing a temporal relation

2) Steps for matching

First we identify if there are temporal related words present in the query. If query contains temporal words then retrieve the posting list of those words from their index file.

For example,

Consider the query Q10 in figure 2. Following steps are involved for searching of relevant documents:

1) Stem the query terms and remove non-relational stop words. Then query Q10 is reduced to:

राम से पहल श्याम घर जाए

2) Retrieve posting lists of query terms excluding relational stop words (see figure 7).

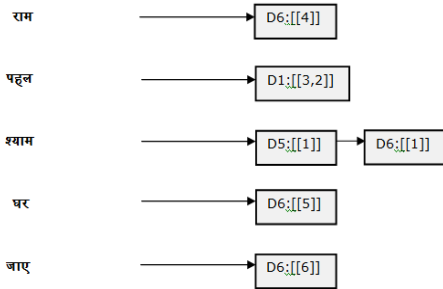
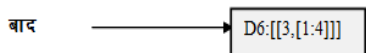


Fig.7. Posting lists of query terms

3) Retrieve the posting list of term opposite of पहले i.e. बाद.



4) Using posting lists, we get the document set containing query terms राम श्याम घर जाए {D6}. From the posting list of बाद relation extracted is श्याम=> बाद<=राम which is opposite of राम =>पहल <= श्याम. So, relation present in D6 is matching with relation present in query.

5) Resultant document set {D6} is retrieved which is the correct answer.

IV. EVALUATION MEASURES

A. **Recall:** It the fraction of the documents that are relevant to the query that are successfully retrieved. [4]

$$Recall = \frac{Number_Relevant_Retrieved}{Number_Relevant}$$

B. **Precision:** It is the fraction of the documents retrieved that are relevant to the user's information need.

$$Precision = \frac{Number_Relevant_Retrieved}{Number_Total_Retrieved}$$

C. **F-measure:** It is the weighted harmonic mean of precision and recall.

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

V. EXPERIMENTAL EVALUATION

We have implemented Information Retrieval System in Java. For experiments, we have used Hindi test collection of about 100 documents extracted from gyannidhi corpus from CDAC. Details of the Hindi corpus are mentioned in table II. Queries on which estimation is done are shown in figure 2. Table III shows normalization of the sample queries of figure 2 for different types of retrieval. Query normalization means removal of stop words and stemming of the remaining words.

We have compared four types of searching: Keyword Based search without Relation Inclusive, Keyword Based search with Relation Inclusive, Concept Based search without Relation Inclusive and Concept Based search with Relation Inclusive.

Keyword based retrieval follow standard normalization in which all the stop words are removed and words are stemmed. In concept based search, query is expanded using synonyms of the words while in relation inclusive search case role relation, spatial relation and temporal relation among words are considered in addition.

A. Estimation of F-measure

Sample queries shown in figure 2 are run on the four environments as mentioned above to get the results. Table IV gives the estimation of F-measure for the four types of searching when they run on same set of queries. From the estimation, it can be seen that F-measure for relation inclusive searching is much higher as compared to that of without relation inclusive. In some cases where synonyms of words are not being used, F-measure is coming out to be same for keyword based searching and concept based searching. For example, for queries Q4, Q6 and Q10 F-measures are 0.57 and 0.80, 0.75 and 0.86, 0.67 and 0.77 respectively for keyword based searching and concept based searching.

TABLE II. DETAILS OF HINDI DOCUMENT CORPUS

	Hindi Document Corpus
No. of Documents	100
No. of index terms	25374
No. of queries	20
Average No. of terms/doc	336

TABLE III. QUERY NORMALIZATION

Query #	Query	Query Normalization			
		Keyword based -Relation Inclusive	Keyword based +Relation Inclusive	Concept based -Relation Inclusive	Concept based +Relation Inclusive
1	दो दिन के बाद बारिश होगी It will be raining after two days	दिन बारिश rain day	दो दिन बाद बारिश rain after two day	दिन बरसात rain day	दो दिन बाद बरसात rain after two day
2	वृक्ष के पास जानवर animal near tree	वृक्ष जानवर animal tree	वृक्ष पास जानवर animal near tree	पेड़ जानवर animal tree	पेड़ पास जानवर animal near tree
3	हिन्दी की पुस्तक टेबल के नीचे है Hindi book is lying below table	हिन्दी पुस्तक टेबल Hindi book table	हिन्दी पुस्तक टेबल नीचे Hindi book below table	हिन्दी किताब टेबल Hindi book table	हिन्दी किताब टेबल नीचे Hindi book below table
4	टेबल पर किताब रखी है book is lying on table	टेबल किताब book table	टेबल पर किताब book on table	टेबल किताब book table	टेबल पर किताब book on table
5	राम निवास के पास खड़ा है Ram is standing near house	राम निवास खड़ा Ram stand house	राम निवास पास खड़ा Ram stand near house	राम घर खड़ा Ram stand home	राम घर पास खड़ा Ram stand near home
6	जानवर ने श्याम को मारा animal killed Shyam	जानवर श्याम मारा animal kill Shyam	जानवर ने श्याम को मारा animal kill Shyam	जानवर श्याम मारा animal kill Shyam	जानवर ने श्याम को मारा animal kill Shyam
7	औरत ने चाकू से फल काटा lady cut fruit with knife	औरत चाकू फल काटा lady cut fruit knife	औरत ने चाकू से फल काटा lady cut fruit knife	महिला चाकू फल काटा lady cut fruit knife	महिला ने चाकू से फल काटा lady cut fruit knife
8	डिब्बे के अंदर पुस्तक book inside box	डिब्बे पुस्तक book box	डिब्बे अंदर पुस्तक book inside box	डिब्बे किताब book box	डिब्बे अंदर किताब book inside box
9	रूम में टी. वी. रखा है T.V. inside room	रूम टी. वी. T.V. room	रूम में टी. वी. T.V. inside room	कमरा टी. वी. T.V. room	कमरा में टी. वी. T.V. inside room
10	राम से पहले श्याम घर जाएगा Shyam will go home before Ram	राम श्याम घर जाए Shyam go home Ram	राम पहले श्याम घर जाए Shyam go home before Ram	राम श्याम घर जाए Shyam go home Ram	राम पहले श्याम घर जाए Shyam go home before Ram

TABLE IV. ESTIMATION OF F-MEASURE

Query #	Keyword based -Relation Inclusive	Keyword based +Relation Inclusive	Concept based -Relation Inclusive	Concept based +Relation Inclusive
1	0.50	0.71	0.67	0.75
2	0.40	0.67	0.59	0.77
3	0.55	0.67	0.63	0.77
4	0.57	0.80	0.57	0.80
5	0.55	0.62	0.57	0.73
6	0.75	0.86	0.75	0.86
7	0.43	0.62	0.57	0.67
8	0.40	0.67	0.60	0.75
9	0.46	0.67	0.62	0.80
10	0.67	0.77	0.67	0.77

Figure 8 gives the graphical representation of the table IV. A graph is plotted between F-measure and the search items to give a comparison of the four search environments. This experiment indicates that system performance is increased in terms of F-measure using relation inclusive searching both in keyword based searching and concept based searching. The performance gains came from query classes which had relations between query terms that could be either case role, spatial or temporal. The experiments showed the benefit of relation based searching by improving precision and recall values. The relation inclusive searching method will give better search performance among documents in English and other similar languages where case roles are represented by spatial/temporal prepositions.

VI. CONCLUSION

In this paper, we have introduced new relation based technique which performs searching of documents using relational, spatial and temporal relations of query terms and gives results better than previously used keyword based and concept based approaches. New indexing technique is used in our method which stores information about case role, spatial and temporal relation along with term position. We have compared four types of searching: Keyword Based search without Relation Inclusive, Keyword Based search with Relation Inclusive, Concept Based search without Relation Inclusive and Concept Based search with Relation Inclusive. For experiments, we have used Hindi test collection extracted from gyannidhi corpus from CDAC. Our proposed searching method gave significant improvement in terms of F-measure. The performance gains came from query classes which had relations between query terms that could be either case role, spatial or temporal. The above relation inclusive searching method will give better search performance among documents

in English and other similar languages where case roles are represented by spatial/temporal prepositions.

References

- [1] A. Bharati, R. Sangal, "Parsing free word order languages in the Paninian framework," in Proceedings of the 31st annual meeting on Association for Computational Linguistics, Association for Computational Linguistics Stroudsburg, PA, USA, doi>10.3115/981574.981589, pp: 105-111, 1993.
- [2] A. Bharati, R. Sangal, "A karaka based approach to parsing of Indian languages," in Proceedings of the 13th conference on Computational linguistics - Volume 3, Association for Computational Linguistics Stroudsburg, PA, USA, ISBN:952-90-2028-7 doi>10.3115/991146.991151, pp: 25-29, 1990.
- [3] G. Cardona, "Panini: His Work and Its Tradition (Vol. 1: Background and Introduction)," Motilal Banarsidas, Delhi, 1988.
- [4] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze. Introduction to Information Retrieval. Cambridge University Press, 2008.
- [5] G. Salton and M. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, 1983.
- [6] Sujoy das, Anurag Seetha, M. Kumar and J.L Rana. "Post Translation Query Expansion using Hindi Word-Net for English-Hindi CLIR System," FIRE 2010, Gandhinagar, Gujrat, India, February 2010.
- [7] Yonggang Qiu and H.P. Frei, "Concept Based Query Expansion," in Proceedings of ACM-SIGIR '93-6/93 /Pittsburgkr, PA, USA, pp: 160-169, 1993.
- [8] Zhiguo Gong, Chan Wa Cheang, and Leong Hou U, "Web Query Expansion by WordNet," In Proceedings of the DEXA 2005, LNCS 3588 166 – 175.
- [9] <http://chitrapurmath.net/sanskrit/supp004.pdf>
- [10] http://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers_in_India
- [11] www.google.co.in
- [12] www.raftaar.in
- [13] www.hinkhoj.com
- [14] "Summary by language size", <http://www.ethnologue.com/statistics/size>

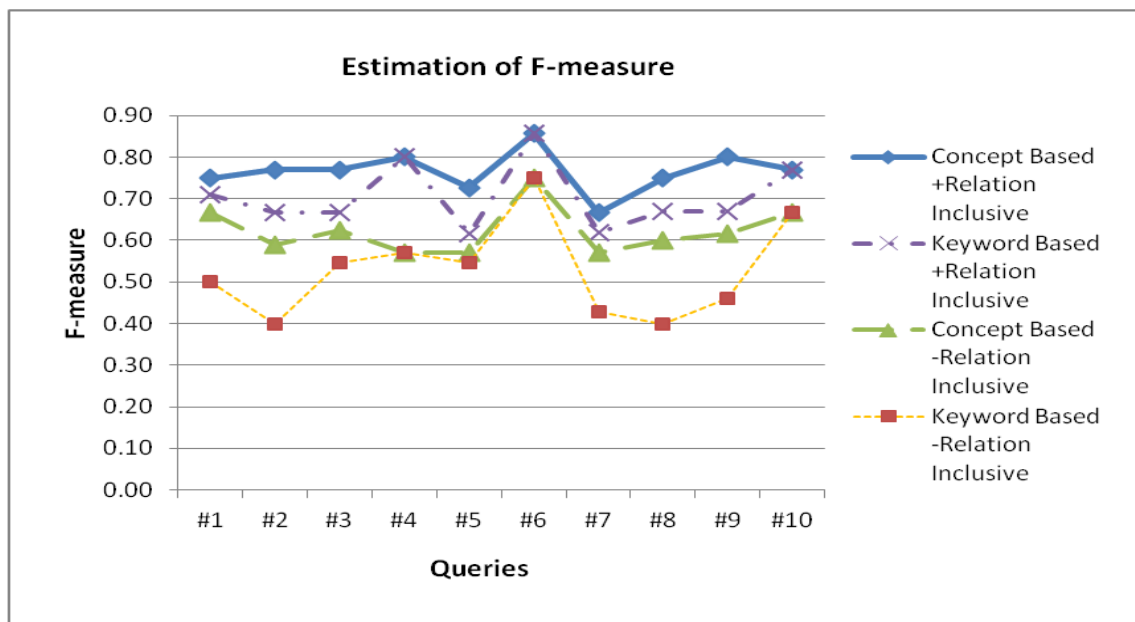


Fig.8. Comparison of F-measure for four types of searching