Journal of Computer Science 4 (9): 762-767, 2008
ISSN 1549-3636

An Entropy Based Method for Removing Web Query Ambiguity in Hindi Language

S.K. Dwivedi and Parul Rastogi
Babasaheb Bhimrao Ambedkar University (A Central University),
Lucknow, Uttar Pradesh, India

Abstract: Problem statement: WSD is core problem of many Natural Language Processing (NLP) tasks; information retrieval is one of them. Information Retrieval in Hindi language also faces the similar problem of WSD. Hindi language is spoken by the major population in India. Natives from the rural area come across the setback of Hindi language information retrieval. WSD is one of them. End users do not understand that how the information retrieval system will remove the ambiguity in the queries. An automatic disambiguation system is required to rectify this problem. Various researchers have worked on it and given solutions. But none of them tried to detect the ambiguity in the query before its disambiguation. **Approach:** We followed entropy based selective query disambiguation approach for Hindi language information retrieval. The approach will identify the ambiguity in the query which will be further disambiguated. The approach is also stimulated by the feature of Google "Did you mean…" for English queries. This study summarizes the ambiguity detection approach as the prior ambiguity detection leads to conserve computation power. **Results:** We applied the selective query approach on the set of fifty queries. In our query set 35% queries were unambiguous. The survey of results concludes that several times even if the query consists of polysemous word, it is detected as unambiguous. **Conclusions/recommendation:** The study concludes that the detection of ambiguity is quiet important as it leads to saving computational time. Followed by ambiguity detection, final disambiguation can be done through human intervention based on google feature.

Key words: Word sense disambiguation, information retrieval, sense ambiguity, polysemous, hindi language, natural language processing

INTRODUCTION

The ambiguity in natural language is considered as the major barrier in language processing applications, especially in information retrieval. Some query terms have a clear cut sense in their query. However some query terms hold ambiguity. The problem also persists with the Hindi language information retrieval as well. Hindi language information retrieval on the web is still in its nascent stage. The number of users who want the information in Hindi language is increasing. This leads to the demand of the Hindi information retrieval on the web. It is the fact that to date Internet is vigorously used in India by the people who are comfortable in English language. The under development of web in Indian regional languages is one of the important reasons behind the limited growth of Internet in India. Indians use 22 official languages and 11 written script forms and among all the languages Hindi language is spoken by the major population of India. About 5% of population understands English as their second language. Hindi is spoken about 30% of the population[4]. This generates the need of the development of the powerful tools for Hindi language information retrieval.

Various search engines are available on the internet as independent search engine sites in English. But very few like (Google, Raftaar and Webkhoj) Hindi language search engines are available. The search engines that support Hindi language search are not able to provide appropriate result for a user query. There are various problems that the search engines face with Hindi language information retrieval. Sense ambiguity is one of the major problems in Information Retrieval on web in Hindi Language. Many words are polysemous in nature. Identifying the appropriate sense of the words in the given context is a difficult job for the search engines. Word sense disambiguation gives solution to the many natural language processing systems including information retrieval.

Corresponding Author: S.K. Dwivedi, Babasaheb Bhimrao Ambedkar University, Lucknow, Uttar Pradesh, India

Sense ambiguity in Hindi language queries can be clearly understood by the given example query "मेहनत का फ़ल (result of hard work)" (in Hindi language) consists of three terms as follows:

Terms	Sense from Hindi WordNet	POS (part of speech)
मेहनत	परिश्रम (hard work)	संज्ञा (Noun)
का	का (of)	कारक (Preposition)
फ़ल	खाने वाला फ़ल (fruit), परिणाम (result), गाँस (upper portion of grass cutting device)	संज्ञा (Noun)

It is unclear from the above mentioned query whether the user is interested in the फ़ल as a fruit, फ़ल as a result or फ़ल in context of device. Here फ़ल is a polysemous word. Before we resolve the ambiguity in query the first step should be the identification of the ambiguity level in the query.

We had tried the approach with the first step of ambiguity detection and finally to resolve query ambiguity we had attempted to use the similar tool "Did you mean……?" of Google for English queries. Though Google also support Hindi language information retrieval but it does not leverage it with the similar facility of "Did you mean…" we had endeavored to apply the same approach for Hindi language queries in which we can confirm from the user the particular sense used in the query. Like "Did you mean फ़ल as a fruit, फ़ल as result or फ़ल in context of cutting device?

The existing Word sense disambiguation tools which map words to their synset can be influenced by the above mentioned motivation to detect the level of ambiguity for each query term. According to our approach if the ambiguity passes the threshold we prompt the user with the two most likely senses. The most likely identified sense can be used for filtration of the documents which do not contain the correct sense.

The WSD approaches used for the English language used WordNet. Our approach used Hindi WordNet[8] which presently incorporates nouns only. So our approach for Hindi Language disambiguation is concerned with nouns only.

The problem statement: The given query is Q which contains one or more query terms as q_1, q_2, $q_3 \ldots q_m$. The query results into the set of relevant document set D. Some query terms are polysemous and have a potential set of senses $S = \{s_1, s_2 \ldots s_n\}$ is for the query Q.

In context of Hindi language Information Retrieval we need to eliminate the कारक (preposition) such as ने, को (to), से (from), के लिए (for), मे (in). and योजक (conjunction) such as या (or), किन्तु (but), परन्तु (but), क्योकि (because), तथा (and), अन्यथा (otherwise). After eliminating these words we have only few keywords left that represent the core query. After the elimination we can detect the ambiguity in query.

The ambiguity is detected in query Q which has polysemous words. We rely on the user input to make the ultimate decision about the possible sense. The user is prompted to select the two most likely senses and selects the correct sense $s_n \in S$.

If the query term q_i is ambiguous the user is allowed to identify the correct intended sense. Further the subsets of results from D that match the intended sense are presented. The disambiguation is related to the resultset rather than the query, because the query is not ambiguous but the result set is ambiguous. It is favorable to identify first the ambiguity in the query. Not all queries are ambiguous in nature. It is necessary to resolve the ambiguity problem to identify queries that can benefit from sense disambiguation.

The process of selecting an intended sense gets tough when no sense has a dominating share in the retrieved result set. If any of the sense dominates the share finding the ambiguity level of the query is quite easy.

MATERIALS AND METHODS

Detecting ambiguity: The focus of the ambiguity detection method is to measure the ambiguity of a query term q_i from a query Q. In general WSD algorithms use probabilistic approach where each sense is tagged with some probability of being correct. The low probability tagging is likely to be ambiguous.

Since our approach is applicable for the information retrieval setup we define the ambiguity of the query in relation to the top k relevant documents for the query. The ambiguity detection is the better option then leading to the disambiguation error. For ex. if there are no documents about the "फ़ल" as a fruit, it will be meaningless to ask the user if they mean "फ़ल" as a खाने वाला फ़ल (fruit).

Following the motivation of[2] the ambiguity of a query term is defined as a function of the senses it takes in the relevant documents. For a query term q_i and a set

of k relevant documents D_k where q_i takes n senses in D_k. They define a maximum likelihood probability distribution p_{qi} over each sense as follows:

$$p_{qi}(s \mid D_k) = \frac{C(s, q_i, D_k)}{\sum_{j=1}^{n} C(s_j, q_i, D_k)} \quad (1)$$

Here we define $C(s, q_i, D_k)$ as the number of times term q_i takes sense s in the set of documents D_k. From this probabilistic sense distribution, we define the ambiguity of a query term as the entropy of its sense distribution. Entropy is the numeric measure of the uncertainty of the outcome:

$$A(q_i, D_k) = -\sum_{j=1}^{n} p_{qi}(S_j \mid D_k) \log p_{qi}(s_j \mid D_k) \quad (2)$$

Finally to detect the ambiguity in the query threshold θ_q is calculated. Threshold is calculated on the basis of entropy of the sense distribution like this:

$$\theta_{qi} = -\sum_{j=1}^{n} p_n(s_j) \log p_n(s_j) \quad (3)$$

If the value of entropy is greater than Threshold or we can say entropy passes a Threshold the query will be an ambiguous query.

Finding most appropriate senses: The Lesk[1] approach which has been modified a bit by the Pushpak Bhattacharya[3] can be followed for finding the two most appropriate senses for the ambiguous words after detecting the ambiguity level of the query. According to Bhattacharya approach:

1. For a polysemous word q_i which needs disambiguation, a set of context words in its surrounding window is collected. Let this collection be C, the context bag
2. For each sense s of q_i, do the following:

(a) Let B be the bag of words obtained from the

- Hypernyms
- Glosses of hypernyms
- Example sentences of hypernyms
- Hyponyms
- Glosses of hypernyms
- Example sentences of hypernyms

(b) Measure the overlap between C and B using the intersection similarity measure

3. Output the sense s_1 and s_2 as the most probable sense which has the maximum overlaps

The idea behind using the intersection similarity measure is to capture the belief that there will be high overlap between the words in the context and the related words found from the Hindi Wordnet[8] lexical and semantic relations and glosses. Now we proceed to the next step of Human intervention.

Human intervention: Human intervention is the next step after finding the most appropriate senses. In this step user will be prompted to select one appropriate sense in a particular context. The user will get now the subset of the relevant document. If the query does not pass the threshold the query will be unambiguous in nature and in that case step 2 and 3 will not be followed.

Related work: Various researchers have studied the effect of ambiguity problem on performance of information retrieval task. According to Sanderson[2] short queries are mostly benefited from the ambiguity resolution. His study showed that disambiguation lead to better performance. Lesk[1] proposed the algorithm for WSD, he also implemented his algorithm on the short text sample and found the good results. With the quite similar approach Pushpak Bhattacharya[3] used his algorithm for the Hindi languaage WSD. His algorithm does not detect the ambiguity in the queries.

Krovetz and Croft[5] studied the relationship between sense mismatch and irrelevant documents. They concluded that the co-occurrence of multiple words interacting within a query naturally performs some element of disambiguation indicating that disambiguation might only be of benefit over short queries.

Weiss[6] showed that ambiguity resolution only lead to the 1% increase in accuracy. The above mentioned all the research deals with the disambiguation of all queries whereas our approach is concerned to the queries where ambiguity is highest. Vogel and Kochher[7] also focused their approach on short sample queries. They suggested disambiguating only those queries where ambiguity is detected. They applied their approach on English queries.

Quantitative Evaluation: Quantitative evaluation of the queries is done on the basis of the above mentioned formula for entropy and threshold.

Hindi language use कारक (preposition), योजक (conjunction). These कारक (preposition) and योजक (conjunction) words will be eliminated from the main query. After eliminating case and conjunction from the queries we are left with the major query terms of the query.

A total of 50 queries are tested on Google search engine and keeping in mind the constraint of limitation of the contents of Hindi language first 20 results are considered for the evaluation. Hindi WordNet[8] is used for sense mapping of the query terms.

Query "मेहनत का फ़ल (result of hard work)" on Google result into 14 relevant documents. After elimination of "का" we left out with the two terms:

- q_1 = मेहनत (hard work) has one sense according to Hindi WordNet
- q_2 = फ़ल (result) has three senses according to Hindi WordNet

The value of probability distribution for मेहनत will be one and Entropy will be 0, hence threshold cannot be calculated.

The set of relevant document set is 14 which means value of k = 14. So the relevant document set is D_k.

The probability distribution of all the senses of query term q2 according to equation 1 is as follows:

- s_1 (□ ल fruit) = 0.2850
- s_2 (परिणाम result) = 0.7140
- s_3 (गाँस upper portion of cutting device) = 0

Entropy is calculated according to the Eq. 2 and the value is 0.2605. Threshold is calculated on the basis of Entropy and it is 1.0745. The value of Entropy is less then the value of Threshold which shows that the uncertainty of the outcome does not passes the threshold. This concludes that this query is not ambiguous.

On evaluation of another query "वर्ण विभेद" on Google we get 18 relevant documents. According to Hindi WordNet we get 3 senses for वर्ण and 1 sense for विबेध:

q_1 = वर्ण (class) and q_2 = विभेद (discrimination)

Here वर्ण is a polysemous word.

The value of probability distribution for विभेद will be one and Entropy will be 0, hence threshold cannot be calculated.

The probability distribution of all the senses of query term q_1 according to equation 1 is as follows:

- s_1 (वर्ण class) = 0.8300
- s_2 (अक्षर alphabet) = 0.1100
- s_3 (रंग color) = 0.0500

Entropy is calculated according to the Eq. 2 and the value is 0.8800. Threshold is calculated on the basis of Entropy and it is 0.1200. The value of entropy is greater then the value of Threshold which shows that the uncertainty of the outcome passes the threshold. This concludes that this query is ambiguous.

The five sample queries are mentioned below:

मेहनत का फ़ल (Result of hard work):

- q1 = मेहनत (संज्ञा/Noun) hard work
- q2 = का (कारक/Preposition) of
- q3 = फ़ल (संज्ञा/Noun) is polysemous

वर्ण विभेद (Class discrimination):

- q1 = वर्ण (संज्ञा/Noun) is polysemous
- q2 = विभेद Discrimination

यशोदा का लाल (Yashoda's son)**:**

Here Yashoda is a name of the lady.

- q1 = यशोदा (संज्ञा/Common noun)
- q2 = का (कारक /Preposition) of
- q3 = लाल (संज्ञा/Noun) is a polysemous word
  - s1 (लाल red color)
  - s2 (पुत्र son)
  - s3 (लाल stone)

नव रस (Nine taste of sentiments)**:**

- q1 = नव (संज्ञा/Noun) is a polysemous word
  - s1 (नया new)
  - s2 (नौ nine)
- q2 = रस (संज्ञा/Noun) is a polysemous word
  - s1 (□ ल का रस juice)
  - s2 (स्राव bodily secretion)
  - s3 (रस several taste of sentiments)

गुलाब की कलम (Rose cutting for planting):

- q1 = गुलाब (संज्ञा/Common Noun)
- q2 = की (कारक/Preposition)
- q3 = कलम (संज्ञा/Noun) is a polysemous word

s1 (लिखने वाला कलम pen)

s2 (तूलिका brush)

s3 (कलम cutting for planting)

The central idea is to consider the distribution of a query term sense in an available relevant document set as discussed earlier. According to the result the term highlighted are ambiguous since the entropy value is greater then threshold. It is evident from the results that even if the query has polysemous word then too it is not considered ambiguous because its entropy is less then Threshold. In this condition we will not prompt the end user to select one appropriate sense.

We used Hindi WordNet [8] as a lexical database for mapping the senses in evaluation work. It is developed at Indian Institute of Technology, Bombay, India. The Hindi WordNet is a system for bringing together different lexical and semantic relations between the Hindi words. It organizes the lexical information in terms of word meanings and can be termed as a lexicon based on psycholinguistic principles.

Entropy and Threshold are used as a measure of the ambiguity detection in the queries. Entropy is solely dependent on the probability distribution of each sense of a particular keyword whereas value of Threshold is dependent on the Entropy itself.

RESULTS

We successfully tested the algorithm specially designed fifty queries (TREC pattern) and a quantitative evaluation of detecting ambiguity for five randomly selected queries is presented in Table 1. The results for the rest of the queries are almost the same.

From the results it is clearly evident that ambiguity detection is quiet important before its disambiguation.

The data in Table 2 clearly shows that out of 50 queries when tested on Google the detection of ambiguity is done successfully in 45 queries. 35% queries were unambiguous even though it consists of ambiguous words.

Our approach successfully identifies the ambiguity in the queries which can further proceed to disambiguation. In general WSD system wastes their computational power in disambiguating the unambiguous query. However early detection of the ambiguity in the queries will save the computational power of the system. It is also evident from the results that many times even if the query consists of polysemous word, it is not ambiguous.

Table 1: Quantitative Evaluation Results

Query	Term (after removal of कारक and योजक)	Senses	Relevant document set	Entropy	Threshold
(Result of hard work)					
मेहनत का	मेहनत	1	14	0.0000	N/A
फ़ल	फ़ल	3	14	0.2605	1.0745
(Class discrimination)					
वर्ण विभेद	वर्ण	3	18	0.8800	0.1200
	विभेद	1	18	0.0000	N/A
(Yashoda's son)					
यशोदा का	यशोदा	1	12	0.0000	N/A
लाल	लाल	5	12	0.9280	0.1883
(Nine taste of sentiments)					
नव रस	नव	2	15	0.2760	0.2376
	रस	11	15	0.1890	0.0630
(Rose cutting for planting)					
गुलाब की	गुलाब	1	16	0.0000	N/A
कलम	कलम	9	16	0.2440	0.0500

Table 2: Overall Results

Total queries	Ambiguity detected	Ambiguous query	Unambiguous query
50	45	30	15

DISCUSSION

The study discussed and summarized the approach for the detection of the ambiguity in the Hindi language queries on the web. The future research will cover the evaluation of the human intervention as well. The human intervention will result into qualitative evaluation of the study.

The approach has certain chances of error as the Hindi WordNet[8] is arbitrarily fine grained. Like in the query "गुलाब की कलम (Rose cutting for planting)" query term "कलम" has 9 senses according to Hindi WordNet, but few senses are hard to distinguish and can be merged. Like sense "पेन (pen)" and "तूलिका (brush)" of keyword "कलम" can be merged. The future study can give the solution by using more robust tools in this context.

So far researchers tried to disambiguate the Hindi language queries like Pushpak Bhattacharya[3]. He used rectified Lesk[1] approach for disambiguation. Lesk used MRD (Machine Readable Dictionaries) whereas Pushpak Bhattacharya[3] rectified his approach and used Hindi WordNet for the disambiguation. He

implemented the Lesk algorithm using the Hindi WordNet lexical semantics for the Hindi languague disambiguation.

Pushpak Bhatacharya[3] had done his experiments for the disambiguation of the Hindi language. Our work is related with the Hindi language information retrieval. In his method he only approached to disambiguate the Hindi language. Besides that the central idea of our work is ambiguity detection.

CONCLUSION

Human intervention in lexical query disambiguation can be an effective tool for information retrieval applications. Detecting the ambiguity using the concept of Entropy and Threshold is found quite successful. Ambiguity resolution improves the performance of the WSD based applications. It reduces the overload on the system by avoiding the useless efforts to disambiguate the unambiguous queries. The ambiguity resolution provides a robust mechanism for presenting results to a user for better conception of the contents of the result set.

REFERENCES

1. Lesk, M., 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone. Proceedings of the 5th Annual International Conference on Systems Documentation, 1986, ACM Press, Toronto, ON, Canada, pp: 24-26. http://portal.acm.org/citation.cfm?id=318728.
2. Sanderson, M., 1994. Word sense disambiguation and information retrieval. Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 3-6, Springer-Verlag New York, Inc., New York, USA., pp: 142-151. http://portal.acm.org/citation.cfm?id=188548&dl=.
3. Bhattacharya, P., M.K. Reddy and P. Pandey, 2004. Hindi Word Sense Disambiguation. www.cse.iitb.ac.in/~pb/papers/HindiWSD.pdf.
4. Burkhart, G.E., S.E. Goodman, A. Mehta and L. Press, 1998. The internet in India: Better times ahead? Commun. ACM., 41: 21-26. http://portal.acm.org/citation.cfm?id=287835.
5. Krovetz, R. and W.B. Croft, 1992. Lexical Ambiguity and information retrieval. ACM Trans. Inform. Syst., 10: 115-141. http://portal.acm.org/citation.cfm?id=146810.
6. Weiss, S.F., 1973. Learning to disambiguate. Inform. Storage Retriev., 9: 33-41. http://eric.ed.gov/ERICWebPortal/custom/portlets/recordDetails/detailmini.jsp?_nfpb=true&_&ERICExtSearch_SearchValue_0=EJ070169&ERICExtSearch_SearchType_0=no&accno=EJ070169.
7. Vogel, A. and S. Kochhar, 2006. Senseable search: Selective query disambiguation. http://sifaka.cs.uiuc.edu/course/498cxz06s/report-as.pdf.
8. Hindi Wordnet from Center for Indian Language Technology Solutions, 2006. IIT Bombay, Mumbai, India http://www.cfilt.iitb.ac.in/wordnet/webhwn/.