Intelligent Query Expansion for the Queries including Numerical Terms

Devendra K. Tayal
Associate Professor, Guru Gobind Singh Indraprastha University, Delhi

Smita Sabharwal
MTech (Information Security), Ambedkar Institute of Advanced Communication, Technology and Research, Geeta Colony, Delhi

Amita Jain
Research Scholar, Jawahar Lal Nehru University, Delhi

Kanika Mittal
MTech (Information Security), Ambedkar Institute of Advanced Communication, Technology and Research, Geeta Colony, Delhi

ABSTRACT
Generally the query input by a user contains terms that do not match those terms which are used to index the majority of the relevant documents. Sometimes the un-retrieved relevant documents are indexed by a different set of terms than those in the query. In order to solve this problem it is necessary to modify the user's query. To do so the researchers have proposed query expansion to help the user to formulate what information is actually needed. For nonnumeric terms researchers purposed many good solutions but as numerical values do not have any synonyms or stemming words, previous approaches were restricted to match the document exactly to the numerical terms that were present in the query. The method presented in this paper searches for the approximate matching of numerical terms also. The method uses fuzzy weighing of query terms with the help of fuzzy triangular membership function.

Keywords
Fuzzy membership function, Numerical terms, Query Expansion, Term-weighting, Information Retrieval

1. INTRODUCTION
For the information retrieval problem, many fuzzy query expansion methods have been studied for a long time. These methods include fuzzy term weighing, fuzzy inference rules, conceptual graphs, generation of fuzzy sets, fuzzy clustering techniques etc. These methods identify terms which are approximately related to query terms for query expansion. Such terms might be synonyms, related concepts, stemming variations or terms which are close to query terms in the text. Unfortunately these methods fail to identify terms which are close to numerical terms present in the query as numerical terms do not have any synonyms or stemming variations. Many times a user query may include numerical terms so in this context the query may be upgraded so that the results are generated over a range of numerical data. This can be done if search documents contain the results from approximate matching of numerical terms. Here, we propose a method through which results will be retrieved ranked on the basis of decreasing level of approximation to the numerical terms mentioned in the query. Our method looks for documents on the basis of a numerical range that is close to the numerical term present in the query. If the numerical value goes far away from the numerical term, its weight decreases and if the numerical value comes close to the numerical term, its weight increases. We make use of fuzzy triangular membership function in our method.

The rest of this paper is organized as follows. In section 2, the related work is mentioned. In section 3, we discuss the basics of Fuzzy triangular Membership function. The next section discusses the basics of Fuzzy Query Term Weighing technique for information retrieval. In section 5, the proposed query expansion for numerical data range is explained. At last we conclude the paper.

2. RELATED WORK
Popular query expansion techniques includes global analysis, association rule based, local analysis, global clustering, user query logs based etc.

In [5], Roberto Navigli, Giuseppe Crisafulli proposed a word sense induction method that first acquires the meanings of query by means of graph based clustering and then clusters the results based on their semantic similarity. Martin Bantista. M. J, Sanchez. D, et al used fuzzy association rules for query refinement. From an initial set of documents text transactions and association rules were constructed which were used to determine the presence of a term in documents with a value between 0 and 1[6]. In [7] K.Lee, W. Bruce Croft and James Allan presented a cluster-based re-sampling method to select better pseudo-relevant documents based on the relevance model. The method used document clusters to find dominant documents for the initial retrieval set, and then repeatedly feed the documents to emphasize the core topics of a query. S Riezler, Y Liu in [8] discussed approaches to process long queries. These methods deployed user query logs to rewrite the query terms into terms from document space using the statistical properties of terms.

Latent semantic indexing, LSI [9] is similar to vector retrieval method in which the dependencies between terms are taken into consideration by modeling all the interrelationships among terms and documents during retrieval. Technology of semantic thesauri proposed by Yufeng Jing and W. Bruce Croft [10] construct a set of words along with a set of relations between these words. However, these techniques lead to large calculations and lowers query efficiency because of word- co-occurrence calculation.

The conceptual graph is used as one of knowledge representation tools. H. Chen, K. J. Lynch proposed a way of regarding concept (keywords or terms) as a node to represent the relationship among concepts [11]. C. Y. Ng proposed an improved algorithm for constructing concept space in 2001 [12]. The algorithm included only strong associations and employed various pruning techniques to avoid computation of weak associations. Query reformulation first expands the initial query with new terms and then reweighs the terms in

the expanded query. Thus, the automatic query reformulation methods improve the initial queries through query expansion and term weighing. However, they do not rely on users to make relevance judgments [13]. They are often based on concept based retrieval [14], language analysis [15], term co-occurrences, PRF [1]. Kim et al. in [18] proposed a query term expansion and reweighting method which considers the term co-occurrence within the feed backed documents.

Jung et al. proposed a terms weighting scheme [16] which considers "absence terms" along with "occurrence terms", in finding the degrees of similarity among document descriptor vectors, in which the "absence terms" means terms which are not present in a specific document and they are provided negative weights rather than assigning zero weights. Klink proposed an automatic reformulation method for improving the original query [17]. As user enters the query in natural language and natural language always contains impreciseness, here we used fuzzy logic. To formulate this concept we use fuzzy triangular membership function as this is the simplest and widely used. Generally user's queries uses natural language and natural language contains lots of impreciseness, so in this paper fuzzy logic is used. We use fuzzy triangular membership function as this is the simplest and widely used.

3. FUZZY TRIANGULAR MEMBERSHIP FUNCTION

In fuzzy sets, each element is mapped to [0, 1] by a membership function. It represents the degree of truth as an extension of valuation. Membership function is given by:

$$\mu_{A:} X \rightarrow [0,1]$$

where, [0, 1] means real numbers between 0 and 1(including 0 and 1) [4].

The triangular membership function is one of the most popular membership functions used. It is given by the following representation of membership values [4].

$$\mu_{(A)}(x) = \begin{cases} 0, & x \le a_1 \\ \frac{x-a_1}{a_2-a_1}, & a_1 < x \le a_2 \\ \frac{a_3-x}{a_3-a_2}, & a_2 < x < a_3 \\ 0, & x \ge a_3 \end{cases} \quad (1)$$

Where, x is a value between a1 and a3, whose membership needs to be calculated between 0 and 1. Fig 1 shows the graphical representation of triangular fuzzy function:

Thus, every numerical value between a1 and a3 can be mapped to a membership value between 0 and 1 using equation 1. From fig 1 it is clear that the maximum membership lies at a2 with value 1.

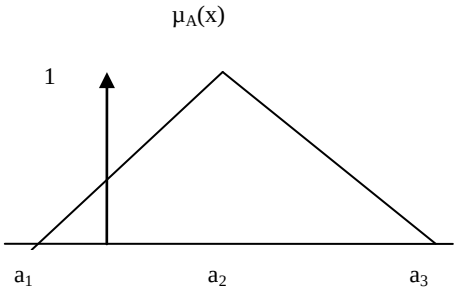

Fig1: Triangular fuzzy function

4. FUZZY WEIGHTED QUERY METHOD

In the weighted query method, the weight wik of a term ti in document dk and the weight wiq of a term ti in the user query Q are calculated as follows[3].

$$w_{ik} = \frac{tf_{ik}}{\max tf_{ik}} \times IDF_i \quad (2)$$

$$w_{iq} = \left(0.5 + 0.5 \times \frac{tf_{iq}}{\max tf_{iq}}\right) \times IDF_i \quad (3)$$

Where, tfik denotes the occurrence frequency of term ti in document dk and tfiq denotes the occurrence frequency of term ti in the user's query Q. IDF is the inverse document frequency of a term ti [1].

The calculation of the degree of similarity S (Q, dk) between the user's query vector Q and the document vector dk is as follows [2]:

$$S(\overline{Q}, \overline{d_k}) = \frac{\sum_{i=1}^{s} 1 - |w_{iq} - w_{ik}|}{s} \quad (4)$$

Where, S (Q, d_k) ∈ [0, 1],

Q = < w_{1q}, w_{2q},…, w_{sq}>, w_{iq} denotes the weight of a term t_i in user query,

d_k = < w_{1k}, w_{2k},…, w_{sk}>, w_{ik} denotes the weight of a term t_i in document d_k,
s denotes total number of terms in user query.

The system translates the original query terms into term weights based on calculated IDF [1] and formula (3) and uses fuzzy rules to infer the weights of additional terms, and then these weights form a query vector Q. The system ranks documents according to their degrees of similarity with respect to the user's query using formula (4) and let the user browse the top n documents.

5. PROPOSED QUERY EXPANSION METHOD

In this section, we propose a new query expansion method that retrieves documents ranked on the basis of decreasing level of approximation to the numerical terms mentioned in the query. We make use of fuzzy triangular membership function to assign weights to the documents having approximate numerical range.

We will use the following notations in our query expansion method:

'q' represent all the terms present in the query,

't_i' represent non-numerical terms present in query,

't_j represent numerical terms present in query,
Thus, $q = t_i + t_j$

v_j = Threshold used to determine numerical range around the numerical term present in the query t_j (v_j may be calculated as 10-15% of the numerical term present in query depending on the application).

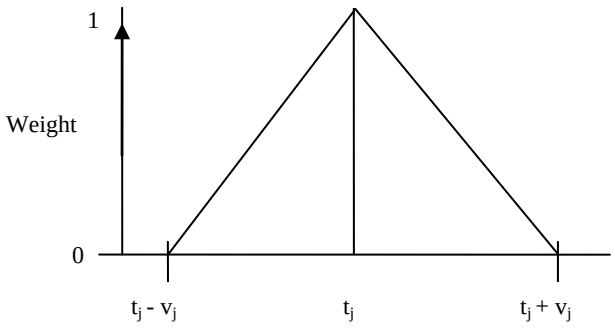

Fig. 2: Fuzzy membership function for term t_j

Fig 2 represents the documents that contains the numerical terms close to t_j will have higher rank over the documents that contains numerical terms far from t_j. Now the query will run for a range of numerical values associated with weights between 0 and 1. The numerical value present in the query (t_j) is assigned highest weight i.e. 1.

The proposed method uses the following algorithm for document retrieval.

5.1 Algorithm

Step1: ***Separating the numerical and non-numerical terms:*** We can make use of a data processing tool to extract the numerical terms (t_j) and non- numerical terms (t_i) from the submitted query.

Step 2: ***For each numerical term in the given query, derive the appropriate numerical range for which the query needs to be run*****:** Once we have the numerical terms we find a range of numerical values using a threshold 'v_j'. Our method will search for all those documents having numerical values that fall in numerical range starting from values $t_j - v_j$ and extending up to values $t_j + v_j$ that are associated with assigned weight.

We would only be considering integer numerical values in this range. Thus we will have ($2v_j + 1$) numerical values for each numerical term present in the query.

Step 3: ***Assigning weights (w_{njkq}) to the numerical range generated in step2:*** Using fuzzy triangular membership function as explained, the numerical terms generated for each numerical term (t_j) in step 1 are assigned weights between 0 and 1. The pseudo- code for step 2 is given below.

Pseudo Code for assigning weights to generated numerical terms

Let 't'$_j$ be a numerical value mentioned in the query.

Taking Threshold 'v_j', the numerical range is from ($t_j - v_j$) to ($t_j + v_j$) which consist of ($2v_j + 1$) values.

a) For each numerical value 'x' less than equal to ($t_j - v_j$)
   Assign weight[x] = 0

b) For each numerical value 'x' greater than ($t_j - v_j$) and less than equal to t_j
   Assign weight[x] = $\frac{x-(t_j - v_j)}{v_j}$

c) For each numerical value 'x' greater than t_j and less than ($t_j + v_j$)
   Assign weight[x] = $\frac{(t_j + v_j)-x}{v_j}$

d) For each numerical value 'x' greater than equal to ($t_j + v_j$)
   Assign weight[x] = 0

ion model is shown in Fig.1.

Now we assigned weights to each numerical value present in the generated numerical range. Considering that each numerical term (t_j) in the query will be associated with a range of 'k' numerical values, we represent the weight of k^{th} element of j^{th} numerical term in the query by $\mathbf{w_{njkq}}$.

Step4: ***Finding relevant documents based on non-numerical component of query and numerical range generated in step2 using fuzzy weighing of query terms*****:** Step 3 gives a range of weighted numerical data values ($\mathbf{w_{njkq}}$). The weights assigned to each numerical term falls between 0 and 1. Now for query expansion we follow the given steps:

a) Assign weights to non- numerical terms (t_i) in the user query using weighted query method as explained in section 4. We represent these non- numerical term weights by $\mathbf{w_{nniq}}$. The weight $\mathbf{w_{nniq}}$ of a non- numeric term t_i in the user query Q is calculated as [3]:

$$w_{nniq} = \left(0.5 + 0.5 \times \frac{tf_{iq}}{\max tf_{iq}}\right) \times IDF_i \quad (5)$$

Where, tf_{iq} denotes the occurrence frequency of term t_i in the user's query Q.
IDF is the inverse document frequency of a term t_i [1].

b) Assign weights to both numerical terms (t_i) and non-numerical terms (t_j) in the documents. The weight $\mathbf{w_{ik}}$ of

term t (numeric as well as non- numeric) in document d_k is calculated as:

$$w_{ik} = \frac{tf_{ik}}{\max tf_{ik}} \times IDF_i \quad (6)$$

Where, tf_{ik} denotes the occurrence frequency of term t in document d_k,
IDF is the inverse document frequency of a term t [1].

c) Calculate the degree of similarity S (Q, dk) between the user's query vector Q and the document vector d_k is as follows:

$$S(\overline{Q}, \overline{d_k}) = \frac{\sum_{i=1}^{s} 1 - |w_{iq} - w_{ik}|}{s} \quad (7)$$

Where, S (Q, d_k) ∈ [0, 1],
Q = < w_{1q}, w_{2q},…, w_{sq}>, w_{iq} denotes the weight of a term t in user query,
w_{iq}= w_{nniq} for a non- numeric term using equation 5,
w_{iq}= w_{njkq} for a numeric term as generated in step 2
d_k = < w_{1k}, w_{2k},…, w_{sk}>, wik denotes the weight of a term t (numeric as well as non- numeric) in document d_k using equation 6,
s denotes total number of terms (numeric as well as non- numeric) in user query.

Fig4 gives the step by step illustration of complete query expansion process.

Now we will explain the proposed method with the help of an example.

5.2 Intelligent Query Expansion Example

We illustrate our algorithm taking an example of a potential buyer looking for mobile phones over internet. Let us consider the search query to be "Mobile Phones around price 10000". The query terms are "Mobile", "Phones", "around", "price", "10000". Applying only term weighing technique would look for these terms, their synonyms and stemming words. However, from the method purposed it is clear that, the user intends to extend his search over a price range around Rs 10000.

Finding range based on threshold vj .We will get (2vj +1) numerical values for each numerical term present in the query. Let vj be (10% of 10000) in this example which is 1000. Thus our numerical search ranges from 9000 (10000-1000) to 11000 (10000+1000) i.e. 2001 numerical values.

Assigning weights to the numerical query range using fuzzy triangular function. The triangular membership graph for this range is shown below

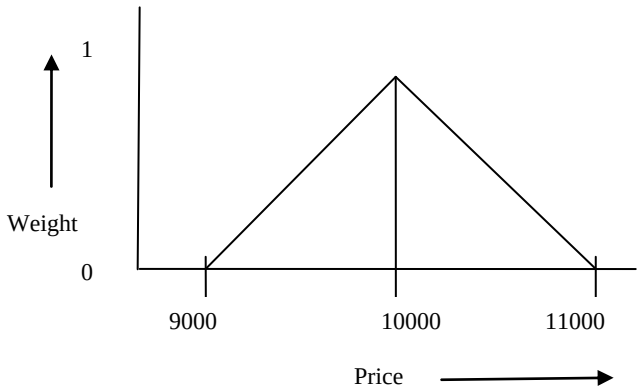

Fig 3: Fuzzy membership function for the example

Thus,

- The documents holding numerical values less than or equal to 9000 will be assigned the smallest weight value as 0.
- The weights assigned to these 2001 numerical values can be calculated as (using equation 1):

weight (9001) = $\frac{9001-9000}{10000-9000} = 0.001$,
weight (9002) = $\frac{9002-9000}{10000-9000} = 0.002$,
.....
.....
.....
weight (9500) = $\frac{9500-9000}{10000-9000} = 0.5$,
.....
.....
.....
weight (10000)=……=1.0
weight (10001) = $\frac{11000-10001}{11000-10000} = 0.999$,
weight (10002) = $\frac{11000-10002}{11000-10000} = 0.998$,
.....
weight (10800) = $\frac{11000-10800}{11000-10000} = 0.2$,
.....
weight (11000)=……= 0.0

Now we combine the numerical and non- numerical terms and expand the query using fuzzy weighted technique. The weights for non- numerical terms are calculated using equation 5. The weights for numerical terms are calculated in step 2. Now using equation 7, the similarity of terms in query and document are calculated and relevant ranked documents are retrieved.

6. CONCLUSION

In this paper, an intelligent method for query expansion is presented that expands the query based on non-numerical as well as numerical terms present in the query. The proposed method is an improvement over traditional query expansion techniques that were restricted to match the document exactly to the numerical terms that were present in the query. This method searches for the approximate matching of numerical terms. So, it is more efficient for queries involving numerical terms. The approximated numerical range associated with weights between 0 and 1 is generated using fuzzy triangular

membership function and then the query is expanded using fuzzy query weighing technique. Thus, the proposed method improves the retrieval accuracy and user satisfaction for the queries having numerical terms.

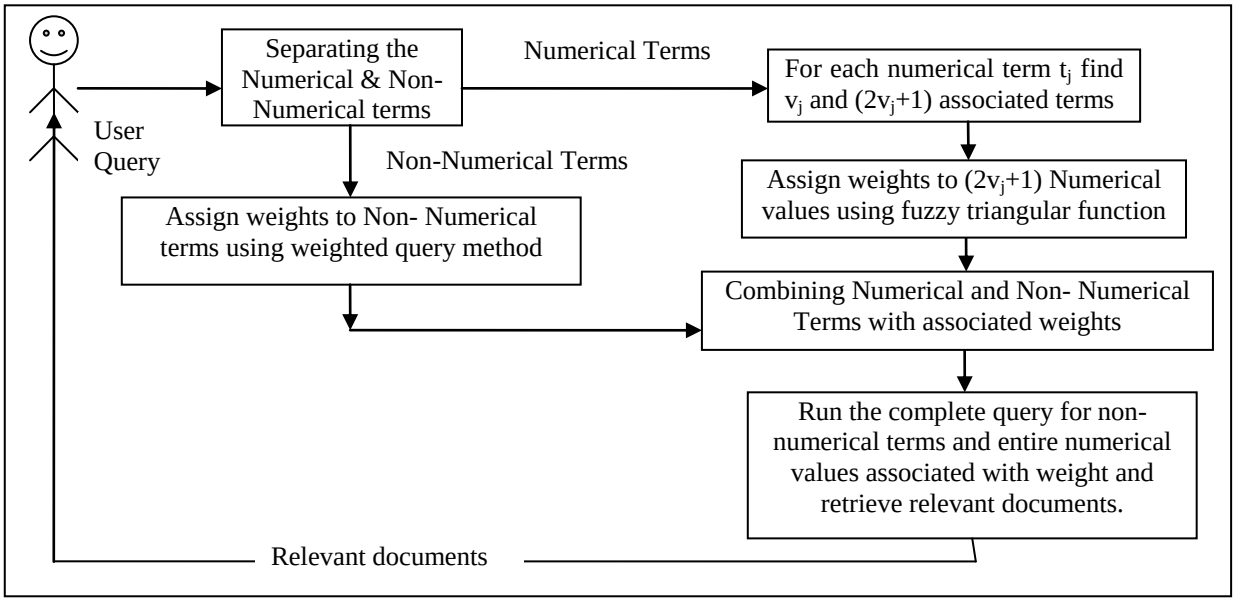

Fig. 4: Proposed Query Expansion Process

7. REFERENCES

[1] Baeza-Yates, R. and Ribeiro-Neto, B.: Modern Information Retrieval. Addison Wesley, New York, 1999.

[2] Horng, Y. J., Chen, S. M. and Lee, C. H.: A new fuzzy information retrieval method based on document terms reweighting techniques, International Journal of Information and Management Sciences, Vol. 14, No. 4, pp.63-82, 2003.

[3] Salton, G.: The Smart Retrieval System - Experiments in Automatic Document Processing. Prentice Hall, Englewood Cliffs, New Jersey, 1971.

[4] Kwang H. Lee: First Course on Fuzzy Theory andApplications, Springer, 2005.

[5] Roberto Navigli, Giuseppe Crisafulli, "Inducing Word Senses to Improve Web Search Result Clustering," in Proc. 2010 Conf. Empirical Methods in Natural Language Processin, Massachusetts, USA, 2010, pp. 116-126.

[6] Martin Bautista, D. Sanchez, J.C. Martinez, J.M. Serrano, M.A. Vila, "Mining Web documents to find additional query terms using fuzzy association rules," Fuzzy Sets and Systems, Vol. 148, No. 1, pp. 85-104, 2004.

[7] Kyung Soon Lee, W. Bruce Croft, James Allan, "A Cluster-Based Resampling Method for pseudo-Relevance Feedback," in Proc. 31th ACM SIGIR conf. Research and Development in Information Retrieval, Singapore, 2008, pp. 235-242.

[8] Stefan Riezler, Yi Liu, "Query Rewriting Using Monolingual Statistical Machine Translation," in Proc. ACL 2010, Uppsala, Sweden, 2010, pp. 569-582.

[9] Dumais. S. T, 1995. "Latent semantic indexing (LSI), TREC-3 report," in Proc. 3rd Text Retrieval Conf. (TREC-3), Maryland, USA, 1995, pp.105-115.

[10] Yufeng Jing, W. Bruce Croft, "An association thesaurus for information retrieval," in Proc. RIAO 94, New York, USA, 1994, pp. 146-160.

[11] H. Chen, K. J. Lynch, "Automatic construction of networks of concepts characterizing document databases," IEEE Transl. J. System Man and Cybernetics, Vol. 22, pp. 885-902, Sep 1992.

[12] Chi Yuen Ng, Joseph Lee, Felix Cheung, Ben Kao, David Cheung, "Efficient Algorithms for Concept Space Construction," in Proc. 5th Pacific-Asia Conf. Advance in Knowledge Discovery and Data Mining, Hong Kong, China, 2001, pp. 99-101.

[13] Y Chang, I Choi, J Choi, M Kim, V.V. Raghavan, Conceptual Retrieval based on Feature Clustering of Documents, 2002.

[14] Qiu, Y. and Frei, H.P. 1993. Concept based query expansion. In Proceedings of the 16th annual international ACM SIGIR conference on Research and Development in Information Retrieval, ACM Press, 160-170.

[15] Bodner, R. and Song, F. 1996. Knowledge-based approaches to query expansion in information retrieval. In McCalla, Advances in Artificial Intelligence, 146-158.

[16] Jung, Y., H. Park and Du, D., An Effective Term Weighting Scheme for Information Retrieval, Computer Science Technical Report TR008, Department of Computer Science, University of Minnesota, Minneapolis, minnesota, pp. 1-15, 2000.

[17] Klink, S. 2001. Query reformulation with collaborative concept-based expansion. In Proceedings of the First International Workshop on Web Document Analysis (WDA2001), Presentation I: Content Extraction and Web Mining, http://www.csc.liv.ac.uk/~wda2001/.

[18] Kim, B.M., Kim, J. Y. and Kim, J., Query term expansion and reweighting using term co-occurrence similarity and fuzzy inference, Proceedings of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference, Vancouver, Canada, Vol. 2, pp. 715-720, 2001.